Encriptació de veu per mescla de subbandes

Pere Salvadó Lloveras

Projecte final de carrera

Encriptació de veu per mescla de subbandes

Enginyeria Tècnica de Telecomunicacions
Especialitat en So i Imatge - EUETIT
Universitat Politècnica de Catalunya

Autor: ***Pere Salvadó Lloveras***
Tutor: ***Ignasi Esquerra Llucià***

Gener 2006

Índex

1 Introducció

En la societat on vivim, regida per el intercanvi massiu de informació, tant per internet, com per mòbils, sense oblidar el telèfon fix ni el fax, ens hem vist obligats a crear sistemes que ens la codifiquin. Sistemes que permetin intercanvi i accés a dades només per aquelles usuaris que tinguin el vist-i-plau per fer-ho.

Un dels camps on més s'utilitza i que més futur té la codificació de dades és internet, degut a la seva massificació i fàcil accessibilitat, tant d'usuaris com de dades. Tothom vol tenir llibertat de moviments per la xarxa, això sí, sense prescindir de la seva privacitat, i per poder-ho garantir en un espai tant transitat, és important utilitzar aquest tipus de sistemes.

L'altre punt important i en gran expansió és la telefonia mòbil. Qui a hores d'ara pot dir que no té mòbil? Probablement molt poca gen. Aquest fenomen, que comporta una gran quantitat d'operacions comercials i relacions a distància, i d'altres, com la televisió per satèl·lit, els canals codificats o la distribució de música per internet, han forçat a la creació de nous sistemes i noves tecnologies encarades a mantenir la privacitat o la restricció d'accés a la informació pertinent. Aquí és on el meu projecte pren sentit. Hi ha varis mètodes per codificar dades, i algun d'ells és a partir del senyal de veu, tractat com a tal i no com a flux de dades. Em centraré a la creació d'un programa que encripti un senyal de veu, o àudio, a partir del seu tractament com a senyal sonor. Concretament empraré un tipus d'encriptació en el que se separa la informació sonora per bandes de freqüència, es redistribueix aquesta informació, i es torna a ajuntar, obtenint així el mateix senyal. Si parlem en termes d'informació, però, ens trobarem amb un senyal ben diferent si ens basem en la intel·ligibilitat. Al posar la informació en un lloc que no li pertoca, aquesta no es perd, però la sonoritat del senyal canvia per complet, quedant així alterada la seva intel·ligibilitat.

Com aquests mètodes han de permetre que l'usuari autoritzat sí que tingui accés a la informació original, també hi ha un procés invers que col·loca la informació on és deguda i permet tornar a tenir el senyal inicial. I la gràcia de la encriptació/desencriptació recau en que aquesta possibilitat de tornar a tenir el senyal original només estarà a l'abast d'aquells que tinguin el sistema de desencriptació adequat, junt amb la clau d'encriptació i el nombre de subbandes en que s'ha treballat en encriptació. Si no és així, el missatge resultant de l'intent de desencriptació no tindrà cap valor, ja que continuarà estant encriptat i al no podrà ser comprès.

1 1 Objectius del Projecte

És un Projecte en el que s'ha tractat d'implementar un programa informàtic que ens encripta la veu, més concretament, es tracta d'un programa encriptador/desencriptador de veu.

L'objectiu d'aquest projecte és implementar en codi Matlab i presentar mitjançant una interfície gràfica aquest enciptador/desencriptador de veu, a fi i efecte d'obtenir-ne un bon funcionament. Entenent per bon funcionament tant la correcta encriptació, desencriptació, i òbviament la recuperació el senyal amb la menor pèrdua possible d'informació. Tot això sense oblidar en cap moment una presentació fàcil d'entendre i d'utilitzar per l'usuari.

També és interessant aconseguir un bon domini del programa Matlab, ja que les seves múltiples aplicacions i gran complexitat proporcionen un bon sistema de tractament de senyals, aplicable en múltiples camp. De gran utilitat en el món de les telecomunicacions.

1 2 Plantejament i procediments

He crear un programa que al entrar-hi un senyal de veu, i definint-li uns paràmetres d'encriptació, com són el nombre de subbandes en que volem fer la encriptació/desencriptació i un codi específic d'encriptació, ens retorni el senyal xifrat. Per aconseguir encriptar-lo el mètode que emprem és el de la mescla de subbandes de freqüència. Consisteix en descompondre el senyal de veu original en un nombre de subbandes determinat i especificat per l'usuari, mesclar-les i seguidament, tornar a ajuntar aquestes subbandes, de tal manera que al tenir el senyal reconstruït no puguem entendre'l, ja que la informació de cada subbanda freqüencial ha estat col·locada en una altra subbanda. Aquesta mescla/reassignació de les subbandes la fem seguint un codi d'encriptació aleatori introduït per l'usuari, que alhora de desencriptar el senyal encriptat, haurà de ser exactament el mateix. El procés de desencriptació és el mateix que el d'encriptació però a la inversa, i per tant, havent fet la mescla a partir d'un nombre aleatori, per desfer aquest mescla caldrà introduir el mateix nombre aleatori.

El programa l'he implementat en llenguatge de programació Matlab. És un llenguatge d'alt nivell i especialment útil si ens endinsem en el tractament de senyals, tant per so com per imatge, ja que ens proporciona eines i funcions ja predefinides que ens proporcionen molta llibertat, així com un sistema de codi molt intuïtiu.

El programa treballa sobre fitxers d'àudio del tipus 'wav' emprant funcions de lectura d'aquest tipus de fitxers ja definides pel programa, i que permeten treballar amb arxius fixes, no pas a temps real. La implementació d'aquest Projecte a Temps Real, és una de les possibles millores que ens hem plantejat al acabar-lo, més endavant comentades.

Finalment he creat una interfície gràfica clara i intel·ligible per la presentació i execució del Programa. En aquesta interfície es permet definir els paràmetres d'encriptació/desencripació al mateix Usuari i alhora pot comprovar

els resultats de la execució del Programa, tant gràficament com sonorament. Ha estat creada amb l'eina Guide del mateix Matlab, aconseguint així una relació programa-interfície òptima.

2 Encriptació

2 1 Introducció a la encriptació

Encriptació és la traducció al català de la paraula anglesa '*encryption*', molt usada en el món de la seguretat en transmissió i accés a dades i informació.

La paraula *encriptar*, i per tant els seus derivats com *encriptació* o *desencriptar*, recentment han estat aprovats per la Secció Filològica de l'Institut d'Estudis Catalans (IEC) i serà introduïda en la segona edició del Diccionari de la Llengua Catalana del IEC, essent aquest el diccionari normatiu de la nostra llengua. Fins ara, ni l'Institut d'Estudis Catalans ni la Real Academia de la Lengua Española no la contemplaven, tot i ser una paraula molt introduïda en la nostra societat, la de la informació.

La definició que s'introduirà serà:

encriptar v. tr. *En informàtica, xifrar basant-se en algorismes matemàtics.*

Entenent per *xifrar*: *escriure emprant una escriptura secreta o xifra.*

Així, d'aquestes definicions en podem extreure que encriptar és el procés mitjançant el qual aconseguim xifrar, codificar, seguint mètodes matemàtics, un text o una informació, aconseguint que no sigui intel·ligible per qui no estigui autoritzat a tenir accés als seus continguts.

Desencriptar ho entendrem com el procés invers, és a dir, conversió de dades encriptades, xifrades, a la seva forma original per tal de recuperar la seva intel·ligibilitat inicial.

La utilització de l'encriptació/desencriptació és tant antiga com l'art de la comunicació inherent a l'èsser humà. Ens hauríem de remuntar a l'època de l'Imperi Romà, per trobar el que es consideren els primers mètodes d'encriptació. Més concretament ens hem de situar en el moment en que l'Emperador Juli Cèsar es va trobar que enviar missatges a les seves tropes que eren al front d'atac era perillós, ja que podien ser interceptats per l'enemic i utilitzats en contra seva. Així, va decidir inventar una codificació per als seus missatges de manera que només les seves tropes, que estaven al corrent d'aquesta codificació, podien entendre'ls, i si mai aquets eren interceptats, l'enemic tampoc podia utilitzar-los per res. El mètode, que informàticament s'anomena algoritme, es tractava del *'n+3'*. La *'n'* fa referència a la lletra de l'alfabet a que es volia fer referència i el tres és el desplaçament respecte aquesta referència que ens marca la posició de la lletra que s'escriu. Així, per referir-se a la lletra A (posició n=1 de l'alfabet), en el missatge escrivien la lletra D (posició 4 = 1+3). Per exemple, si volien que el receptor entengués la paraula CESAR, en el missatge escrivien la paraula FHVDU. En aquell moment, en que mai s'havien utilitzat aquest tipus de mètodes, va ser una gran solució i ajuda als projectes d'expansió de l'Imperi Romà, però actualment els algoritmes utilitzats per encriptar missatges, informació o qualsevol tipus de dades que mereixin ser encriptades, ofereixen una complexitat infinitament superior al *'n+3'* de Juli Cèsar.

En qualsevol procés d'encriptació, ha d'haver-hi un procés de desencriptació posterior, ja que si s'encripta és perquè hi ha una informació que només interessa que algú autoritzat pugui accedir-hi, i per tant, encara que la gent autoritzada sigui molt poca, és necessari tenir un procés invers a la encriptació que permeti recuperar la informació inicial. En el cas esmentat anteriorment, que l'encriptació era *'n+3'*, el procés de desencriptació, que ha de ser sempre l'invers, era *'m-3'*, sent ara 'm' la lletra encriptada.

També és important tenir en compte que no només és necessari tenir accés al procés de desencriptació, si no també tenir accés a la clau d'encriptació, en els casos que sigui necessària. La clau d'encriptació, com el

seu nom indica, és un paràmetre (nombre, paraula, lletra,…) que alguns sistemes d'encriptació empren alhora de xifrar la informació, i si l'usuari que vol recuperar aquesta informació no té la clau adequada no podrà recuperar-la correctament, la informació continuarà encriptada.

El bon funcionament d'un sistema d'encriptació, basant-nos en la fiabilitat del manteniment de la privacitat de la informació, recau en la seva complexitat algorísmica. A major complexitat algorísmica, menor probabilitat de que les dades puguin ser obtingudes per algú no autoritzat, és a dir, més complicat serà que algú sense la clau d'encriptació ni el procés de desencriptació adequats desxifri la informació confidencial.

2 2 Tipus d'encriptació

Un cop definit el concepte d'encriptació i desencriptació, cal endinsar-nos-hi més, ja que hi ha molts tipus d'encriptació.

Basant-nos en la seva complexitat algorísmica, tenim els encriptadors simples i els complexos. Entre els simples podem trobar-ne que es basin en la substitució de lletres per números, en la ja esmentada rotació de les lletres de l'alfabet (permutació) o en el sistema que veurem en aquest projecte de reordenació de les bandes freqüencials d'un senyal d'àudio. I entre els encriptadors complexes, que s'implementen a partir de complicats algoritmes informàtics/matemàtics que treballen sobre la informació bit a bit, podem trobar-ne que canviïn per exemple els bits de dades a senyals numèriques i les tractin. Aquests últims, treballen sempre sobre informació digitalitzada.

2 2 1 Encriptadors de veu

Com ja s'ha comentat, els mètodes d'encriptació de veu o àudio no són excessivament complexes, si s'implementen a partir del tractament del senyal com a so i no com a dades digitals. Bàsicament hi ha tres sistemes i les seves possibles combinacions.

- Els que alteren l'amplitud del senyal
- Els que n'alteren la freqüència
- Es que en pertorben el temps del senyal

Exposats de menor a major seguretat

2 2 1 1 Modificació de l'amplitud

Són sistemes que modifiquen l'amplitud del senyal i hi afegeixen un to modulat en freqüència amb la amplitud a xifrar del senyal. Per desxifrar el senyal encriptat es desmodula la amplitud desitjada del to alhora que es filtra el to portador. És un sistema poc segur.

2 2 1 2 Modificació temporal del senyal

Aquest mètode és necessari implementar-lo digitalment i treballar amb els senyals d'àudio digitalitzats. Així el senyal està expressat en forma de cadena de valors, temporalment ordenats. El sistema el que fa és agrupar les dades per blocs temporals i els permuta dins el vector. L'agrupament es fa per varis blocs per segon i la permutació segueix una seqüència determinada matemàticament per una clau d'encriptació.

Aquest és un mètode especialment útil per a l'encriptació a temps real, tot i que degut als càlculs i permutacions, el senyal pateix un retard que el fa poc apta per a comunicacions ràpides, i alhora necessita una perfecta sincronització entre l'emissor i el receptor. Sempre tenint en compte que tant l'un com l'altre han de disposar de l'encriptador i del desencriptador, així com de la clau utilitzada.

2 2 1 3 Modificació de la freqüència

Hi ha varis mètodes d'encriptació d'àudio a partir de la modificació del senyal en freqüència:

- Inversió de l'espectre:

Com el seu nom indica, s'inverteix l'espectre del senyal, quedant les freqüències greus com a agudes i viceversa. Si l'ample de banda del senyal és petit, entre 3 i 4 KHz, no s'entén res del senyal xifrat sense desencriptar-lo, és a dir, té molt poca intel·ligibilitat, i en canvi, si parlem d'amples de banda elevats, el senyal gaudirà d'una major qualitat en detriment de la seguretat.

Per realitzar aquesta inversió de freqüències amb processos digitals el que es fa és invertir el valor de les mostres parells. El senyal ha d'haver estat digitalitzat, i per tant, l'hem de tenim en forma de cadena de mostres.

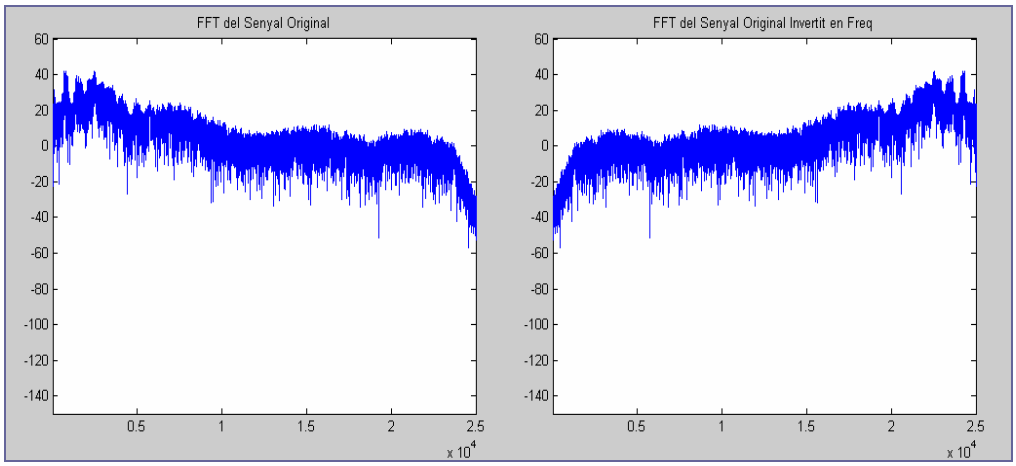

Figura2.1

En les gràfiques de la *Figura2.1* es pot observar clarament el funcionament del procés. S'ha agafat un senyal de veu masculina en format d'arxiu d'àudio '*wav*', ja digitalitzat, i se li ha invertit tots els valors de les mostres parells. Les dues gràfiques són de la FFT dels dos casos, l'original i l'encriptat. La FFT d'un senyal ens mostra la seva distribució espectral. Mirant aquestes dues gràfiques podem observar com a la primera (senyal original) tenim molt pes informatiu a baixes freqüències (dreta de la gràfica), i poca a altes (extrem esquerra), i a la segona podem comprovar que els senyal és exactament el mateix però invertit, quedant la informació d'aguts en els greus i viceversa. Així, si escoltem el senyal, ens sona com a fregit, degut a la gran quantitat d'informació d'aguts, i a la manca de greus.

Aquest és el sistema més estès d'ecriptació de veu, però alhora el menys segur, ja que per fer-ho no hi ha possibles claus i qualsevol que sàpiga aquest sistema, pot desencriptar un senyal.

Un exemple molt familiar per a tots d'utilització d'aquest sistema per encriptar senyal d'àudio és el *Canal+*. Això inicialment, ja que en vistes de la gran facilitat per desencriptar el senyal, la xarxa de pirateria d'aquests sistemes de televisió digital estava molt estesa. Actualment i després de tres canvis de sistemes de codificació de les plataformes digitals de televisió espanyoles, la plataforma *Digital+* utilitza un sistema de codificació de *Nagravisión*, conegut

amb el nom de *Cardmagedon*. Un sistema d'encriptació digital que no entrarem a analitzar, ja que no segueix els paràmetres acústics, sinó informàtics.

- Inversió i partició de l'espectre:

Aquest és un sistema que ens afegeix un grau de dificultat al que acabem d'explicar. Consisteix en dividir l'espectre del senyal invertit en dues parts i desplaçar-ne una, col·locant l'altra en la part buida deguda al desplaçament.

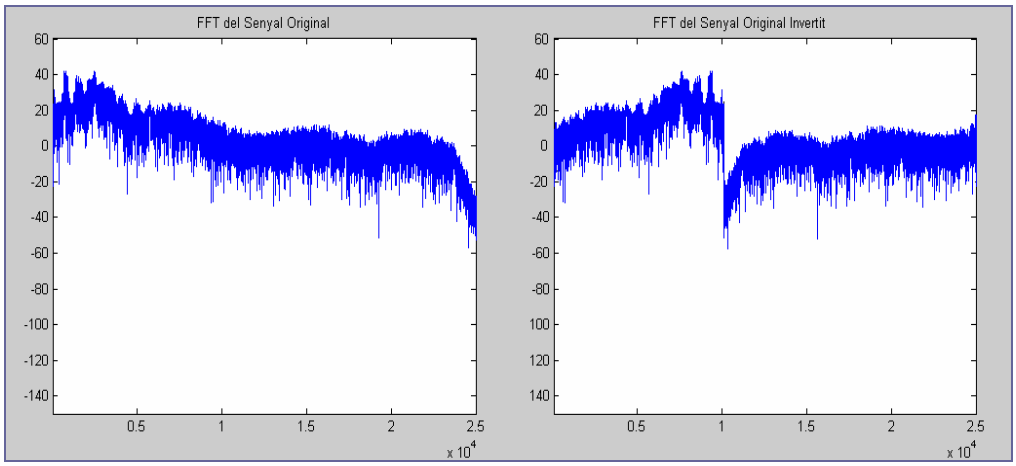

Figura2.2

- Fragmentació i inversió espectral i mescla de les subbandes:

Aquest és el sistema en que es basa el Projecte, concretament en una de les seves variants. Té un funcionament bàsic que després admet diferents procediments.

El concepte bàsic és la partició de l'espectre del senyal en vàries subbandes, mesclar aquestes subbandes i recomposar-lo. Obtenint així un espectre completament diferent.

A partir d'aquí, es pot complementar de diverses maneres, augmentant-ne la seva fiabilitat a mesura que hi afegim detalls.

La primera opció és la de fer la descomposició, la inversió o no d'algunes subbandes i la posterior recomposició sempre igual. Seguint un procés predefinit i establert, sense la contribució de paràmetres externs com podrien ser una clau.

La segona opció seria el mateix procés acabat d'explicar però ara la permutació de les subbandes i la seva inversió o no, es decideixen seguint un cicle repetitiu. Aquest procés és més segur, ja que l'algoritme ja no són unes assignacions fixes, entrem en un procés cíclic molt més segur informàticament parlant.

El tercer és el procés en el que s'aprofundeix en aquest projecte, però en el que s'entrà en detall més endavant. Ara en veurem el funcionament bàsic. Compleix les mateixes bases que els altres dos, però la complicació recau en la utilització d'una clau d'encriptació aleatòria alhora de la permutació de les subbandes. L'existència de la clau d'encriptació farà que sense aquesta sigui realment complicada l'obtenció del senyal original, tenint en compte l'enorme nombre de possibles combinacions de permutació que ens permet fer la descomposició del senyal en subbandes. Aquesta clau és un nombre o lletra que ens dona un valor numèric concret, i és a partir d'aquest valor, i mitjançant càlculs matemàtics, com s'arriba a una permutació de les bandes aleatòria. Tant aleatòria com el valor de la clau. Així, per a la recomposició del senyal serà necessària aquesta clau i el seu valor, ja que per poder dur a terme el procés inversament idèntic al d'encriptació, és aquest valor i la seva inversa el que ens permetran que la permutació segueixi també el camí al revés. Per tant, en aquest punt ja s'entra en sistemes d'encriptació prou elaborats, que no poden desxifrar-se amb quatre càlculs fixes, degut principalment a l'aleatorietat de la clau d'encriptació.

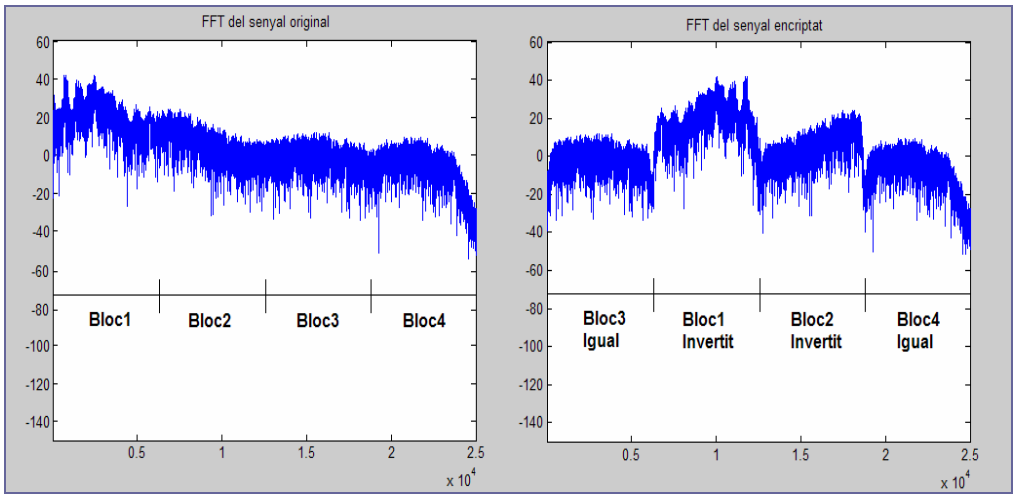

Figura2.3

En la *Figura2.3* podem observar dues gràfiques, l'espectre del senyal original i la del senyal encriptat. S'observa la partició en subblocs del senyal original, i tot seguit l'alteració que ha patit aquest senyal al ser sotmès a un sistema d'encriptació com el que acabem de definir. Podem observar clarament quina permutació ha sofert cada subbloc, i també quins s'han invertit i quins no. Tot això degut als complicats processos matemàtics que aquest encriptació comporta i a la contribució del valor aleatori de la clau d'encriptació.

Executant el mateix sistema d'encriptació amb el mateix senyal de veu, però amb una clau d'encriptació diferent, es pot veure en *Figura2.4* com el valor aleatori d'aquest clau és de gran influència.

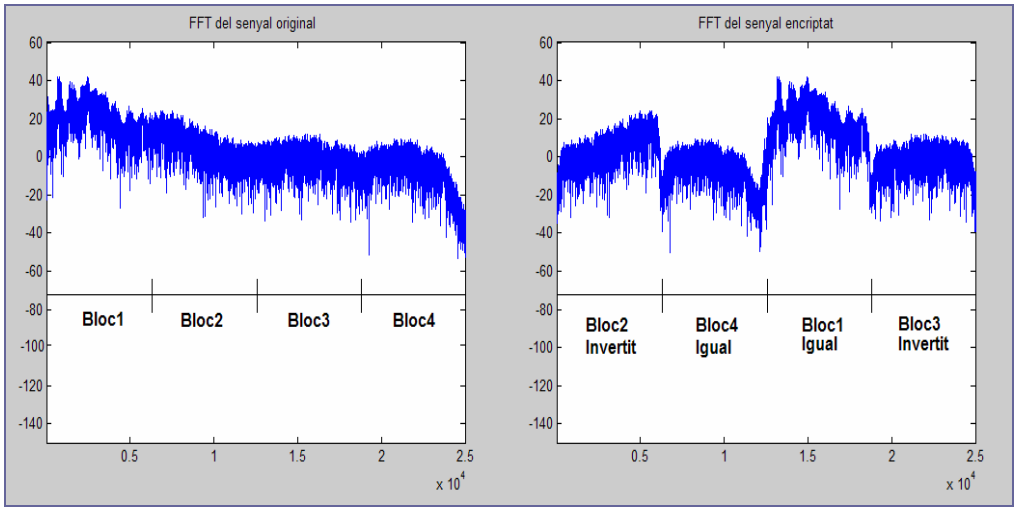

Figura2.4

2 2 2 Encriptadors Complexos

Entenem per encriptadors complexos aquells basats en algoritmes matemàtics que treballen sobre informació digital (cadenes de valors), independentment del tipus de dades que aquesta informació digital representa.

S'implementen amb algoritmes matemàtics, el que fa que encara que la complexitat dels càlculs no sigui molt alta, el seu desxiframent sigui bastant més complicat, sobretot per el fet que són càlculs duts a terme a partir de la informació digitalitzada, i com a tal, sigui quin sigui el format original de la informació representada, no seguiran paràmetres característics d'aquesta. Parlant de senyals d'àudio, es pot decidir encriptar seguint mètodes com els descrits anteriorment, que es basen en les seves característiques físiques (freqüència, amplitud, ...), o digitalitzar el senyal i encriptar-lo tractant les dades com a simples cadenes de valors, sense fixar-se en què és el que estan representant.

Algoritmes d'encriptació d'alta complexitat matemàtica, i per tant, de molt difícil desxiframent, n'estem rodejats. Avui en dia qualsevol tipus d'informació que es transmet passa per un procés d'encriptació/desencriptació.

Es pot diferenciar entre dos tipus de sistemes s'encriptació complexes, els de Clau Secreta (o simètrics) i els de Clau Pública (o asimètrics).

2 2 2 1 Sistemes de Clau Secreta (o Sistemes Simètrics)

Són sistemes on apareix la funció *E*, anomenada funció d'encriptació, que depèn de dos paràmetres. El primer paràmetre és la clau *K* i el segon és la informació *M* que es pretén mantenir secreta. La privacitat s'aconsegueix a través de la encriptació del missatge *M* utilitzant la clau *K*, és a dir, mitjançant la

determinació de *C=E(K,M)*. Quan es parla de text pla, ens estem referint a *M*, i ens referim *C* com a text xifrat, encriptat.

El text xifrat pot ser emmagatzemat o enviat a través d'algun canal de transmissió, sigui segur o no, a una altra entitat. En ambdós casos, per recuperar la informació original, el text pla, és estrictament necessari conèixer la clau *K* i utilitzar una funció de desencriptació *D* que també depèn dels mateixos dos paràmetres de que depèn la funció d'encriptació *E*, de manera que ara *D(K,C)=M* per a qualsevol cas de *C=E(K,M)*.

En aquests sistemes, la idea és que la funció d'encriptació sigui tal que sigui infactible determinar res útil respecte el text pla a partir del text xifrat si no es té accés a la clau.

D'acord amb el tipus d'operacions amb que s'utilitzin els mecanismes d'encriptació es parla de sistemes d'encriptació del blocs o per fluxe.

· Sistemes d'encriptació per blocs:

Són aquells ens que les dades es divideixen en blocs de caràcters, normalment del mateix tamany. Posteriorment, cada bloc s'encripta utilitzant una mateixa transformació que ha estat donada per una mateix clau.

Entre els sistemes més importants, en destaquen:

· DES (*Data Encryption Standard*) i les seves variants
· AES (*Advanced Encryption Standard*) – Rijndael (l'algoritme base del AES)
· FEAL (*Fast Data Encryption Agorithm*)
· IDEA (*International Data Encryption Algorithm*)
· Safer (*Secure and Fast Encryption Routine*)
· RC5 (*Rivest's Code 5*) i RC6 (*Rivest's Code 6*)

· Sistemes d'encriptació per fluxe:

Són aquells sistemes en que els caràcters (típicament es parla de caràcters binaris) s'encripten d'un en un donant lloc a una transformació d'encriptació que varia en el temps.

Entre els sistemes més coneguts destaquen:

· *One-Time Pad*
· *Feedback Shift Register*
· RC4 (*Rivest's Code 4*)
· SEAL (*Software-optimized Encryption Algorithm*)

2 2 2 2 Sistemes de Clau Pública (o Sistemes Asimètrics)

Són sistemes basats en claus d'encriptació de la forma *K=(P,S)* que consta de dues parts, una pública *P* a la que tothom hi pot tenir accés i una altra secreta *S*. La seva utilitat recau en que permeten que una entitat emissora, posem pel cas B, enviï informació confidencial a una altra de receptora, diguem A, a través d'un canal de comunicacions insegur. En aquests sistemes, cadascuna de les entitats que desitja establir aquesta comunicació i rebre o enviar certa informació confidencial, crea una clau *K=(P,S)*, fa pública la part *P* i manté en reserva la part *S*. A més, en aquests sistemes també existeix la funció *E*, anomenada funció d'encriptació, que depèn de dos paràmetres. En aquest cas el primer paràmetre és una clau *P* i el segon és la informació *M* que es pretén mantenir en secret. La privacitat s'aconsegueix a través de la encriptació, per part de l'emissor, del missatge *M* mitjançant la clau *P*, és a dir, mitjançant la determinació de *C=E(P,M)*, d'on *M* és el text pla, l'original, i *C* el text xifrat, encriptat.

El text xifrat s'envia a través d'algun canal de comunicacions insegur al receptor A aquí correspon la clau pública *P* (i que en conseqüència coneix la clau secreta *S*, associada a *P*). En els dos casos, per recuperar el missatge original és necessari conèixer la clau *K* i utilitzar una funció de desencriptació *D*, que també depèn de dos paràmetres, la clau secreta S i el text xifrat *C*, essent *D(S,C)=M* la seva expressió, per a qualsevol *C=E(P,M)*.

La seguretat d'aquests sistemes recau en que tot i ser P de domini públic, és computacionalment infactible determinar el text pla *M* a partir de *C=E(P,M)*.

El gran avantatge dels sistemes de clau pública és que el maneig de les claus és significativament més senzilla. En efecte, si és possible autentificar claus, aleshores la distribució de claus es pot fer de manera més senzilla ja que no són necessaris mètodes segurs i fiables de distribució com en els sistemes de claus privada. Aquest maneig també es veu substancialment simplificat gràcies a que cada entitat (receptora o emissora) només necessita generar i resguardar una sola clau, la seva clau secreta *S*.

El desavantatge principal d'aquests sistemes respecte els de clau privada és la seva lentitud, són significativament més lents. Per això els sistemes de clau pública s'utilitzen principalment per transmetre i establir claus que subsegüentment són utilitzades per a criptosistemes de clau privada.

Entre els sistemes de clau pública més coneguts en destaquen:

· RSA
· Criptosistemes de *Corves El·líptiques*
· Criptosistema de *Rabin*
· ElGamal
· McEliece

2 2 3 Evolució dels estàndards d'encriptació

En l'actualitat, el algoritmes xifradors utilitzats per els principals estàndards d'encriptació que hem esmentat són una combinació de simples que hem explicat al principi, els de substitució i els de permutació. Com ja s'ha comentat, l'objectiu bàsic de l'anomenada 'confusió' que crea la combinació d'aquests algoritmes simples és el d'amagar la relació existent entre les dades originals, les dades encriptades i la clau s'encriptació. La major part dels sistemes d'encriptació es basen en això, en varies capes de substitucions i permutacions, estructures anomenades SPN (*Substitution-Permutation Networks*). Les SPN o reds de substitució-permutació són un xifrat iteratiu, és a dir, consisteixen en aplicar un nombre concret de rondes o voltes de substitucions i permutacions a cadascun dels bolcs o dades, depenent de com es tractin les dades.

Com ja s'ha vist, un dels principals algoritmes de xifrat per bolcs és el DES (*Data Encryption Standard*) i podríem dir que és el sistema d'encriptació de clau privada més utilitzat arreu del mon, i molt especialment en l'àmbit financer i bancari. El DES xifra i desxifra blocs de 64 bit i els sotmeten a 16 rondes, amb una clau d'encriptació de 54 bits (56 bits reals i 8 bits de paritat)

El maig de 1973 la *National Boureau of Standards*, conegut actualment com *National Institute of Standards and Technology*, el NITS, del govern nord-americà va sol·licitar a la comunitat científica sistemes d'encriptació simètrics, amb vistes a adoptar un estàndard que pogués ser construït en massa i que proporcionés les màximes garanties de seguretat. No va ser fins el 1975 que es va fer oficial l'algoritme escollit i que el govern dels EEUU va decidir, després de grans discrepàncies sobre si era un algoritme realment segur, adoptar-lo com a estàndard per a comunicacions no classificades. S'especulava que podia ser que aquest algoritme tingués propietats algebraiques, les quals permetrien desxifrar fàcilment les dades encriptades, i que eren mantingudes en secret per el govern Nord-Americà, però la veritat és que mai s'han pogut verificar aquests rumors. També s'ha de dir que s'esperava que aquest estàndard tingués una

vida d'entre 10 i 15 anys, però amb el temps s'ha comprovat que no, que ha superat i amb escreix les expectatives. Per sorpresa de tothom al 1990 va poder suportar atacs de sistemes d'encriptació diferencials, acabats de descobrir per E.Biham i A.Shamir, fet que fa pensar que els investigador de IBM creadors del DES ja coneixien aquest tipus d'atacs 20 anys abans que la resta del mon però que van mantenir-ho en secret fins la descoberta de Biham i Shamir. La principal crítica del DES es refereix al tamany real de la clau d'encriptació de 56 bits, la qual és realment massa petita per poder garantir una bona seguretat. Sense anar més enllà, si ens fixem en el seu predecessor, el sistema *Lucifer* també de IBM, constava d'una clau de 128 bits. En aquest sentit, s'han anat fent intents d'atacs al DES, al 1977 Diffie i Hellman van suggerir la construcció d'un xip que comprovés 10^6 claus per segon, aconseguint desxifrar la clau en un dia si tinguéssim una màquina composta amb 10^6 xips d'aquest tipus. Això sí, es va preveure que tindria un cost de 20.000.000 euros, així que no es va dur a terme mai. Al juliol de 1998 la empresa *Electronic Frontier Foundation* va construir l'ordinador '*DES Cracker*' que contenia 1536 xips amb capacitat per buscar 88 billons de claus per segon, i al gener del 1999 van unir-se el *DES-Cracker* junt amb el *DES-Challenge-III* (fabricat per la companyia RSA) i junt amb 100.000 ordinadors més a través d'internet, i finalment van aconseguir trencar el sistema DES en 22 hores i 15 minuts, havent compovat més de 245 billons de claus per segon.

En la actualitat el DES continua sent utilitzat arreu. Hi ha moltes companyies que prefereixen mantenir aquest sistema d'ecriptació, precisament per el fet que ha estat capaç d'aguantar durant més de 20 anys, i opten per utilitzar-ne variants com ara el TDES (actua tres vegades el DES, amb tres claus) evitant així el risc que suposa canviar de sistema.

Tot i això, sembla que el DES i totes les seves variants tenen els dies contats, ja que des de 2003 el NIST va convertir l'estàndard AES (*Advanced Encryption Data*) en l'estàndard oficial nord-americà, fet que provoca una axtensió generalitzada d'aquest estàndard arreu del planeta. Serà només en

certs sectors del mercat en que és obligatori l'ús del DES per normativa o per compatibilitats el que farà que no arribi a desaparèixer del tot.

L'AES ja hem vist que és un sistema de xifrat simètric, com el DES, però treballa amb clau d'encriptació de 128,192 i 256 bits, així com blocs de treball de 128 bits de tamany.

Tot i no definir-lo com l'estàndard oficial nord-americà fins el 2003, aquest va ser creat definitivament el 2000. Concretament la publicació oficial de l'algoritme definitiu del AES va ser el 2 d'octubre d'aquell any.

A diferència de la creació de DES dues dècades abans, aquest cop es va optar per una convocatòria pública oberta a tot el món, proposada el gener de 1997. En les bases de la convocatòria es van especificar els requisits mínims d'acceptació:

- L'algoritme havia de ser públic, disponible gratuïtament i que s'ajustés al requisits de la política de patents de l'*Institut Nacional Americà d'Estàndards*.
- Havia de ser un algoritme de xifrat en bloc simètric.
- Havia de ser dissenyat de tal manera que tingués flexibilitat a l'hora de variar la longitud de la clau depenent de les necessitats.
- Havien de poder suportar xifrats amb longituds de bloc de 128 bits i longituds de clau de 128,192 i 256 bits.
- S'havia de poder implementar tant en hardware com en software.

També es van especificar els criteris d'evaluació:

- Seguretat
- Eficiència computacional i requisits de memòria
- Adequació hardware i software

- Simplicitat de disseny i flexibilitat
- Requisits de llicència.

El termini de presentació d'algoritmes participants acabava el 15 de juny del 1998 i aleshores començaria el procés de selecció, consistent en tres trobades a diferents punts del mon on els algoritmes eren sotmesos a probes, comentaris i revisions.

De les 21 propostes inicials, només 15 complien les exigències de convocatòria i després de les dues primeres trobades, a l'agost del 1998 i al març del 1999, van quedar 5 candidats finalistes presents a la tercera i última trobada convocada per l'abril de 2000 a New York:

· MARS
· RC6
· RIJNDAEL
· SERPENT
· TWOFISH

Finalment, el 2 d'octubre del 2000 el NIST va fer públic el resultat de la convocatòria i per sorpresa de tothom, el guanyador va ser l'algoritme RIJNDAEL, un algoritme belga creat per Vincent Rijmen i Joan Daemen, vencent així a algoritmes nord-americans i a criptòlegs de reputada fama mundial, com Bruce Shneier (qui va liderar l'equip de ceació de l'algoritme TWOFISH) o Ronald Rivest (dissenyador del RC6 i de l'antic RSA).

Els motius exposats pel NIST per haver seleccionat aquest algoritme van ser:

- Molt bona combinació seguretat-velocitat-eficiència, tant en memòria com en portes lògiques
- Senzillesa
- Flexibilitat

Una de les virtuts de l'AES, és a dir de l'algoritme Rijndael, respecte a l'anterior, el DES, és l'elegància, des del punt de vista matemàtic, en la seva descripció. També s'ha de tenir en compte que l'algoritme en si, permet xifrar blocs de 128, 192 i 256 bits, amb claus també de 128, 192 i 256 bits, però que és l'estàndard AES, que només permet el xifrat i desxifrat de blocs de 128 bits, i que limita les grans possibilitats d'aquest algoritme. Potser per això s'espera que aquest algoritme tingui una llarga vida.

3 Encriptador/desencriptador de veu per mescla de subbandes

Immersió al sistema que ja s'ha introduït anteriorment, aquell en que l'encriptació del senyal es fa seguint les seves característiques físiques, concretament, fragmentant l'espectre i invertint i mesclant-ne els subblocs.

El sistema d'encriptació que s'ha creat podríem col·locar-lo en el grup dels sistemes simètrics, ja que la privacitat es basa en una clau secreta que només coneixeran els usuaris a qui se'ls permeti tenir accés a la informació xifrada. Més concretament, seria una variant d'aquests sistemes, ja que no depèn d'una clau secreta d'encriptació, si no de dues. En aquest sistema encriptador l'usuari pot definir dos paràmetres que faran que la encriptació sigui una o una altra. Estem parlant de la clau d'encriptació secreta, que anomenarem *K*, i que ens determina quina permutació seguiran les subbandes en que s'ha descompost el senyal original, i del nombre de subbandes en que fem aquesta descomposició. Si el procés d'encriptació el fem descomponent el senyal original en un nombre de subbandes que anomenarem *n* i permutant questes subbandes amb la clau d'encriptació *K*, i després intentem recuperar el senyal original a partir de l'encriptat, però fent el procés de desencriptació descomposant el senyal encriptat en una altre nombre de subbandes, diferent a *n*, o bé utilitzant una clau diferent a *K*, no obtindrem pas el resultat desitjat. N'obtindrem un senyal que encara continuarà encriptat. Així, l'encriptador implementat segueix l'estructura de $Y=E(X,K,n)$, entenent per *X* el senyal original, *K* la clau d'encriptació (secreta), *n* el nombre de subbandes (també secret), *E* la funció d'encriptació i *Y* el senyal encriptat.

La recuperació del senyal original a partir de l'encriptat, haurà de seguir una estructura molt similar, i es farà mitjançant un sistema desencriptador que compleix la igualtat $X=D(Y,K,n)$, sent *D* la funció de desencriptació i *X*, *Y*, *K* i *n* els mateixos paràmetres que en la encriptació.

Així, queda palès que la privacitat del sistema recau en la clau secreta d'encriptació *K* i en nombre de subbandes *n* en que s'ha fet la encriptació. Sense el coneixement d'aquests dos paràmetres, és realment molt difícil, per no dir quasi impossible, poder desxifrar un senyal d'àudio que ha estat encriptat amb aquest sistema.

Es pot considerar un sistema d'encriptació/desencriptació de veu prou segur.

Després de fer una introducció a la veu i el seu tractament digital, descompondrem l'explicació en tres blocs principals.

El primer serà la *idea bàsica*, on entrarem a explicar la idea del sistema encriptador/desencriptador, fixant-nos en la encriptació i en la desencriptació per separat.

La segona serà on entrarem a explicar en detall la descomposició en subbandes, i per tant els seus components més destacats: els filtres, la delmació i interpolació, i la influència de la freqüència de mostreig del senyal original alhora d'escollir el nombre de subbandes.

Finalment, entrarem a explicar en detall la mescla de les subbandes, el perquè, les matrius de commutació i la influència de la clau d'encriptació. Tot, tant per la encriptació com per la desencriptació.

3 1 Introducció a la veu i el seu tractament digital

Per entrar a veure la veu i el seu tractament digital, primer cal explicar què és el so i les ones sonores.

3 1 1 El so i les ones sonores

El so el podem definir com una vibració mecànica que es propaga a través d'un medi homogeni i elàstic que és capaç de produir una sensació auditiva. A diferència de les ones electromagnètiques, el so no es pot propagar per el buit, sempre és necessari un medi, per exemple l'aire. La manera en que es propaga la vibració amb que obtenim el so és el que anomenem '*ona sonor*a'. És un tipus d'ona longitudinal, és a dir, la direcció de la excitació que ha provocat la vibració coincideix amb la direcció de propagació.

La ona sonora és producte de la variació de pressió del medi transmissor degut a una vibració mecànica. Aquesta vibració produeix una compressió i una posterior dilatació de les partícules que trobem en el medi de propagació, i això sent un procés periòdic ens crea una ona de pressió que va oscil·lant periòdicament. És aquesta l'anomenada ona sonora.

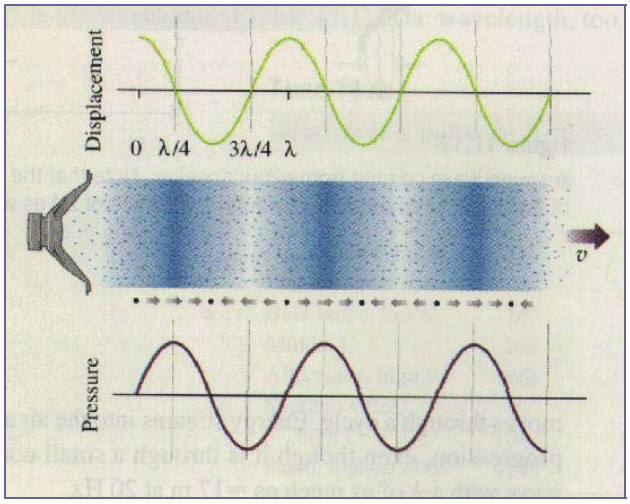

Figura3.1

En la *Figura3.1* veiem la relació entre la compressió i dilatació de les partícules de l'aire producte de la vibració de la membrana de l'altaveu i les característiques físiques de la ona sonora.

Així, les característiques queden definides per tres paràmetres. L'amplitud o pressió sonora, la longitud d'ona i en conseqüència la freqüència.

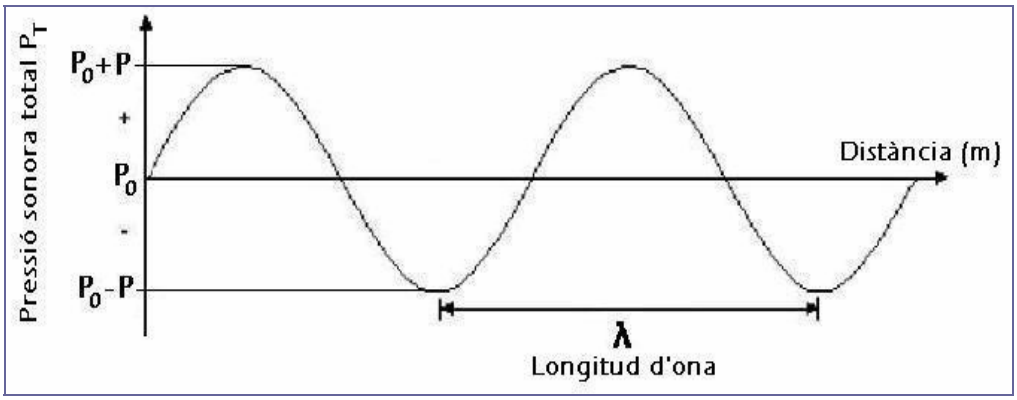

Figura3.2

El nivell de pressió sonora d'una ona és la mesura de la pressió a que es veuen sotmeses les partícules del medi de propagació arran de la seva compressió i dilatació producte de la vibració mecànica. A major compressió major pressió, i a major dilatació menor pressió. Així, el nivell de pressió màxim ens determina el volum, junt amb altres paràmetres psicoacústics, del senyal que s'està escoltant. La longitud d'ona és la distància entre dos punt de la ona que es troben el mateix estat de vibració, i és el que ens determina la freqüència del senyal escoltat, és a dir, el to. Es mesura en metres.

La relació entre la longitud d'ona (λ) i la freqüència (f) és la següent:

$$\lambda = {}^{c}/_{f}$$

On λ és la longitud d'ona expressada en metres, c és la velocitat de propagació del so, i f és la freqüència. Recordem que c depèn del medi de propagació i les seves característiques, i fixem uns 345m/s com a valor estàndard de propagació en l'aire.

RELACIÓ ENTRE FREQÜÈNCIES I LONGITUDS D'ONA

LONGITUDS D'ONA - metres						
17,6	0,7	0,345	0,173	0,086	0,035	0,0176
20	500	1000	2000	4000	10000	20000
FREQÜÈNCIES - Hz						

Figura3.3

A partir de la longitud d'ona d'un senyal en determinem la seva freqüència, i són aquests dos paràmetres els que determinen la tonalitat del so, el que coneixem per notes, així com aguts, greus, mitjos, etc.

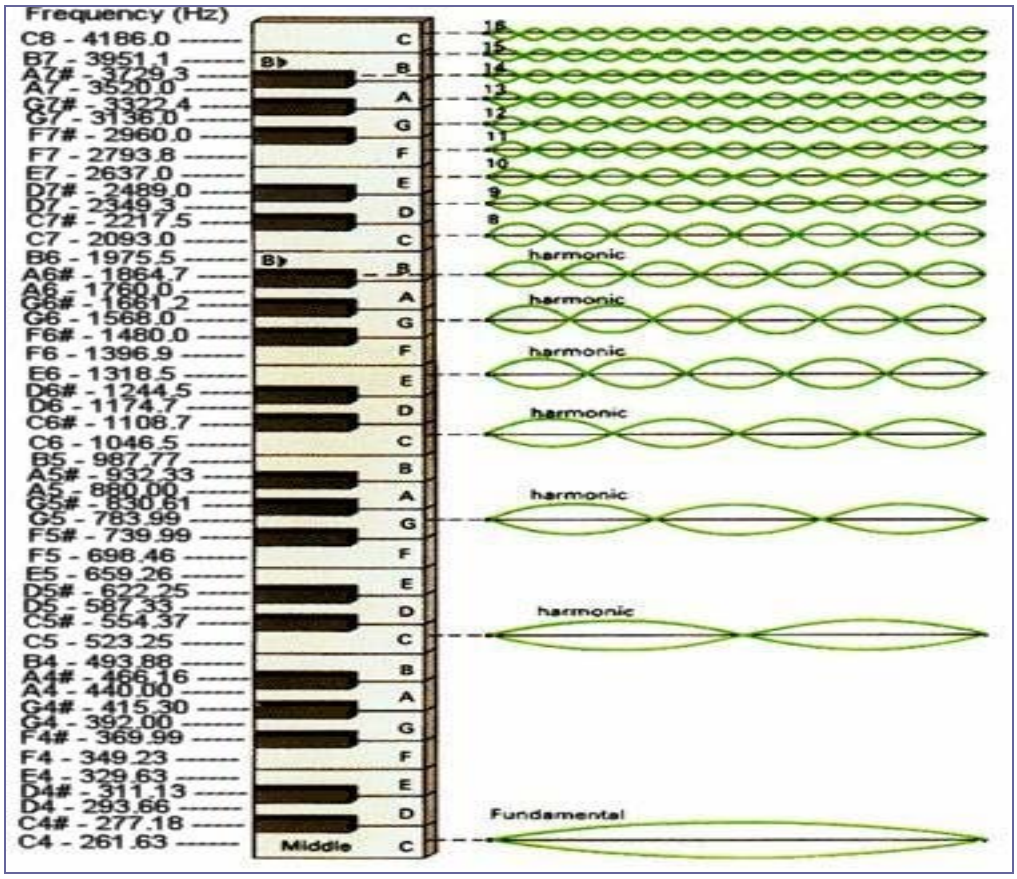

Figura3.4

Entre la *Figura3.3* i la *Figura3.4* podem apreciar la relació entre notes o tons, freqüències que representen i les pertinents longituds d'ona. En el segon cas, observant les tecles d'un piano podem veure quina relació hi ha entre les notes i els seus harmònics ja que només estan representades les longituds d'ona d'un to fonamental i els seus harmònics. Els harmònics són les freqüències múltiples d'un to fonamental. Múltiples per un nombre enter. Per exemple, 2kHz

i 3kHz són el primer i el segon harmònic si prenem com a freqüència fonamental 1kHz.

3 1 2 La veu

La veu i per tant la parla és el so que en surt de la boca del éssers humans. És produït per la vibració de les cordes vocals, combinat amb la resta de components que formen el tracte vocal. Aquest, el podem descompondre en tres formants: la laringe i la faringe són el tercer formant, la cavitat nasal el segon, i la cavitat bucal el formant principal.

Concretament, el que considerem veu, és un flux d'aire provinent dels pulmons i la tràquea, modulat de forma periòdica per la vibració de les cordes vocals creant-ne així una ona periòdica, i per tant un to, que més endavant és modulat tant en freqüència com en amplitud per els formants que componen el tracte vocal. Aquests tenen un comportament com de ressonadors, es comporten com a filtres dinàmics (laringe i faringe no) i així modulen el so tant en freqüència com en amplitud.

La veu, la parla, la podem descompondre en dos tipus de sons, els sons sonors i els sons sords. Dins els sonors trobem les vocals i algunes consonants (B, D i G), i dins els sords la resta de consonants. Són els primers, els sonors, els que són producte del tracte vocal, en canvi els sords són producte de l'obstrucció i el posterior alliberament de l'aire que altres parts de la boca provoquen (llengua, llavis,...).

La veu té un ample de banda d'uns 4Khz, concretament la podem situar entre els 115 Hz i els 4Khz aproximadament. Depenent sempre del locutor, trobem que la freqüència fonamental, que és la freqüència més baixa capaç de reproduir aquest locutor, no sempre és la mateixa. A trets generals, podem

establir que la freqüència fonamental d'un home cau en els 115Hz, la d'una dona en uns 220Hz i la d'un nen/nena al voltant dels 300Hz.

Pel que fa a la intel·ligibilitat de la parla i al nivell de la veu, fixant-nos en els sons sonors i sords trobem que els sonors (vocals) són els que tenen una major contribució al nivell de la veu, i en canvi són els sords (les consonants) els que defineixen el grau d'intel·ligibilitat del missatge.

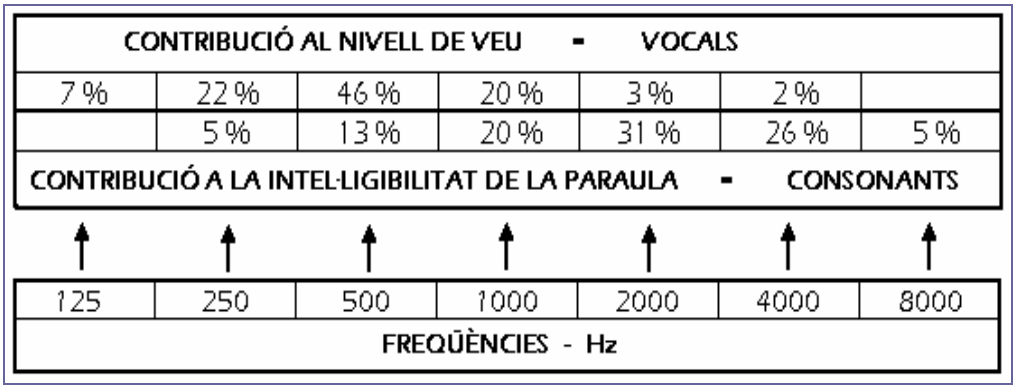

Figura3.5

La *Figura3.5* ens representa la contribució per bandes freqüencials dels sons sords o sonors pel que fa a la intel·ligibilitat del missatge i al seu nivell, en el cas d'una veu d'home. Veiem que la contribució al nivell de la veu recau sobretot al voltant dels 500Hz, baixes freqüències, i en canvi, la màxima intel·ligibilitat la trobem a la vora dels 2KHz, altes freqüències. Així doncs, queda palès que en un senyal de veu seran les vocals i les baixes freqüències que ens marcaran el nivell de la veu, deixant que siguin les consonants i altes freqüències el que ens defineixin el nivell d'intel·ligibilitat del missatge oral. Aquest és un exemple per a una veu d'home adult, i hem de dir que per el cas de la dona, la distribució segueix la mateixa estructura, però tot a una freqüència una mica més alta. En la veu de dona, la zona de màxima intel·ligibilitat la trobem aproximadament a 4KHz.

Per poder implementar l'encriptador/desencriptador de veu per mescla de subbandes informàticament necessitem treballar amb arxius del tipus 'wav', i això significa que caldrà que la veu que vulguem encriptar hagi passat

prèviament per un procés de digitalització i s'hagi emmagatzemat dins un arxiu 'wav'.

3 1 3 Tractament digital de la veu i l'àudio

La digitalització del so ja s'utilitza en el punt en que es vol emmagatzemar, sigui en arxius del tipus informàtics o directament en suports d'àudio per ser escoltat.

El tipus més comú de gravació d'àudio digital és la modulació de codi de polsos (PCM – *Pulse Code Modulation*). És el tipus utilitzat per a la gravació de discs compactes (CD) i per la majoria dels arxius del tipus 'wav', els que utilitzarem en el nostre encriptador/desencriptador.

Alhora de la gravació del tipus PCM, un micròfon converteix la variació de la pressió de l'aire (les ones sonores) en un voltatge variable (senyal elèctric) i un convertidor analògic a digital ens digitalitza aquest voltatge variable. Aquest procés de digitalització es du a terme a partir del mostreig o quantificació del senyal elèctric. Es van fent mesures del voltatge variable del senyal en un interval regular de temps, anomenat '*període de mostreig*' (*Tm*), i s'emmagatzemen digitalment. Per exemple, en la gravació dels discs compactes, es prenen exactament 44.100 mostres cada segon. Aquesta taxa de mostres per segon és el que entenem per '*freqüència o velocitat de mostreig*' (*Fm*), que és l'invers del període *Tm*. En aquest punt cal tenir en compte el '*teorema de Nyquist*', que estableix que per evitar problemes d'*aliasing* cal que la freqüència de mostreig sigui com a mínim del doble de l'ample de banda del senyal. Per això, la discretització de veu s'acostuma a fer amb una *Fm* de 8KHz (recordem que la veu té un ample de banda d'uns 4KHz), i els senyals de música s'acostumen a mostrejar amb una velocitat superior als 40KHz (44'1KHZ, 48KHz,…) ja que l'oïda humana, i per tant qualsevol tipus de

música audible, està 'limitada' entre els 20Hz i els 20KHz (ample de banda d'uns 20KHz aproximadament). Un cop feta la discretització temporal del senyal, ara cal emmagatzemar aquests valors variables de l'amplitud del senyal digitalment i per això cal discretitzar-los, amb un nombre prou elevat de bits. Concretament, en telefonia digital es fa amb 8 bits, però amb senyals més complexos que els de veu, com poden ser senyals de música amb qualitat de CD, s'utilitzen 16 bits, o fins a 20 bits per a mescles digitals de CD.

Així, les mostres PCM són aquesta cadena de mostres obtinguts amb la digitalització del senyal, i que el representen temporalment. La estructura d'aquesta cadena de mostres depèn de si ens trobem amb un senyal '*estèreo*' o un senyal '*mono*'. Si el senyal és estèreo, que vol dir que tenim informació diferent per la oïda esquerra que per la dreta, les mostres s'emmagatzemen alternament, és a dir, es van intercalant les mostres d'un canal amb les de l'altre. Seguint aquesta estructura: *esquerra1, dreta1, esquerra2, dreta2, esquerra3, dreta3,...* Si en canvi, tenim un senyal en 'mono', on la informació és idèntica per una oïda o per l'altra, les mostres simplement d'emmagatzemen l'una rera l'altra.

En el cas del nostre encriptador/desencriptador, treballarà sempre amb arxius del tipus 'wav'. El format 'wav' (*waveform audio file*) és un format d'arxiu originari de *Microsoft Windows 3.1*. Era el format d'emmagatzematge d'àudio més usat, però domèsticament està sent desbancat per el famós '*mp3*', degut a que aquest últim té un tamany molt menor al 'wav', això sí, en detriment d'una compressió que depenent de les característiques provoca una pèrdua considerable d'informació. El 'wav' en contrari, és un format que ocupa molt, però que alhora té una qualitat d'àudio molt millor, gairebé sense pèrdues, però sempre depenent de la compressió i gravació, així com del mostreig usat.

Concretament, els arxius 'wav' són simples emmagatzemadors, és a dir, no és pas un format de compressió d'àudio, si no un format d'arxiu que permet emmagatzemar un senyal d'àudio que ha estat comprimit digitalment seguint

algun dels tipus de compressió que pot suportar. El tipus de compressió més usat en els fitxers 'wav' és el format PCM que ja hem vist, però també podem trobar-ne que continguin senyals comprimits segons la *Llei-A*, segons la *Lei-μ* o en format *ADPCM*.

Per l'encriptador/desencriptador el tipus de compressió del senyal d'àudio que conté l'arxiu 'wav' que s'encriptarà/desencriptarà és irrellevant ja que el sistema està implementat íntegrament amb el programa Matlab el qual té una funció ja predefinida anomenada '*wavread*', que com el seu nom indica, ens llegeix arxius del tipus 'wav', independentment del tipus de compressió que duguin dintre.

La representació dels senyals que ens farà el Matlab no serà la mateixa si parlem d'un senyal 'estèreo' o d'un 'mono'. Si es tracta d'un senyal estèreo ens el presenta en dos canals separats, concretament ens crea dos vectors, l'un amb la informació, en l'ordre temporal del senyal, del canal esquerra i l'altre amb la del dret, i col·locats 'en paral·lel', tractant-ho com una matriu de dues files per tantes columnes com mostres té cada canal, facilitant-ne així molt la maniobrabilitat. Si en canvi es tracta d'un arxiu mono ens col·loca les mostres en un sol vector, seguint també l'ordre temporal del senyal.

3 2 Idea bàsica de l'encriptador/desencriptador

El funcionament bàsic de l'encriptador/desencriptador es podria explicar en conjunt, però s'explicarà en dos passos, per poder veure clarament quina és la diferència entre el procés d'encriptació i el de desencriptació.

3 2 1 Encriptador:

Com ja s'ha explicat anteriorment, la base d'aquest sistema d'encriptació és la descomposició de l'espectre del senyal de veu o àudio en subbandes. Això és gràcies a les característiques del so, que ens permeten una representació del senyal del so al llarg del camp freqüencial, i la seva partició en subblocs (subbandes freqüencials). Aquesta partició de l'espectre es du a terme mitjançant uns processos de filtratge del senyal, concretament, és el filtratge pas-alt i pas-baix del mateix senyal, obtenint-ne una descomposició en dues subbandes, les altes i les baixes freqüències respectivament. A partir d'aquesta partició en la qual ja s'ha obtingut el senyal original partit en dos, i en funció del nombre de subbandes en que es vulgui fer la descomposició, repetirem el procés de doble filtratge per a cada part, obtenint-ne ara dos blocs més de cada bloc. El sistema creat permet decidir si l'usuari vol descompondre el senyal en 2, 4, 8 o 16 subbandes, depenent també de la freqüència de mostreig del senyal que es vol encriptar, ja que per a freqüències de mostreig molt baixes, no val la pena fer una descomposició en moltes subbandes. Com ja s'ha comentat, la possibilitat de decidir el nombre de subbandes ens dóna un grau més de dificultat alhora de desxifrar el senyal encriptat.

Anàlisi:

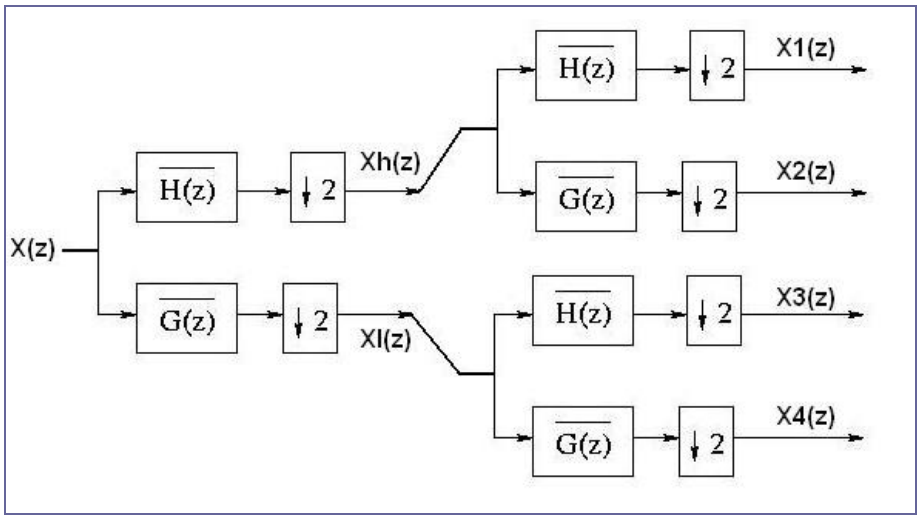

Figura3.6

En la expressió de la *Figura3.6* s'aprecia el procés a que és sotmès el senyal d'àudio en el cas de fer una descomposició en quatre subbandes freqüencials, procés anomenat ANÀLISI. On *H(z)* és un filtre passa-alt i *G(z)* és un filtre passa-baix. El procés requereix la delmació per poder mantenir el nombre de mostres del senyal original, ja que cada filtre es dedica única i exclusivament a filtrar, i per tant, manté el nombre de mostres del senyal d'entrada. Així com mantenir cada part filtrada situada on li pertoca de l'espectre. Són temes que s'explicaran més endavant.

Un cop obtinguda cada subbanda el que es fa és redistribuir-les, mesclar-les, col·locant la informació d'una banda en el lloc d'una altra, fent-ho per totes les subbandes. Aquesta permutació de les subbandes es fa mitjançant una matriu de commutació que ha estat creada a partir de la clau d'encriptació secreta i aleatòria que ha introduït l'usuari al seu gust i depenent del nombre de subbandes, també decidit per l'usuari. En aquest procés és on ens quedaran definides les característiques del procés d'encriptació, aquelles que el de desencriptació necessitarà saber si es vol recuperar els senyal original a partir de l'encriptat obtingut al finalitzar aquest sistema encriptador. Aquest procés de permutació de les subbandes és el que s'anomena de '*mescla*'.

Mescla:

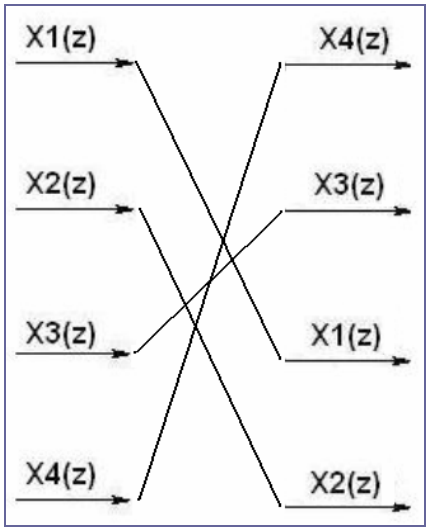

Figura3.7

En aquest representació de la *Figura3.7* s'aprecien les permutacions que patirien en un cas hipotètic les subbandes en que ha estat descompost el senyal anteriorment. En aquest cas tindríem la informació de baixes freqüències (*X4*) en el lloc de les altes, les altes (*X1*) a les mitges-baixes, les mitges-altes a les baixes i les mitges-baixeses a les mitges-altess.

Per acabar amb el procés d'encriptació ara el que cal és tornar a ajuntar les subbandes. En aquest procés, anomenat de síntesi, sotmetem cada subbanda a una interpolació i al seu posterior filtratge. En aquest punt és on acabarà de quedar definida la posició de cada subbanda, per quin canal freqüencial ens apareixerà en el senyal encriptat, depenent de si és filtrada per un filtre passa-alt o per un passa-baix.

Síntesi:

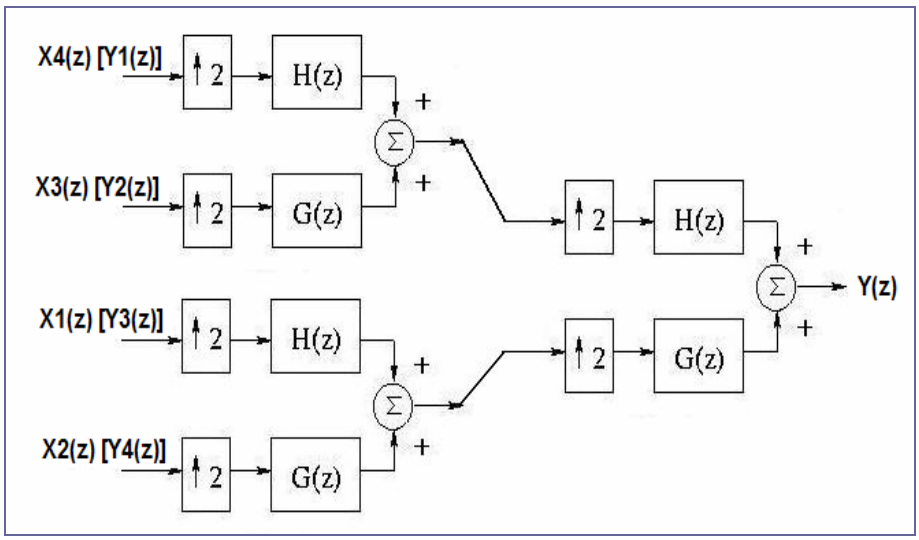

Figura3.8

Un cop passats aquests tres processos que componen el sistema d'encriptació, el senyal *Y(z)* que s'ha obtingut és el senyal original *X(z)* encriptat.

3 2 2 Desencriptador

El sistema desencriptador segueix exactament la mateixa estructura que acabem de descriure, anàlisi, mescla i síntesi, però amb la permutació de les subbandes invertida. És aquí on recau la diferència entre un procés i l'altre. Per això és tant important saber el nombre de subbandes en que s'ha descompost el senyal en la encriptació i tenir accés a la clau d'encriptació, ja que ara es tornen a utilitzar, però inversament. La matriu de commutació que es crearà en aquest procés ha de ser la inversa a la del procés d'encriptació, i per tant, és necessari saber en quantes subbandes s'ha treballat en la encriptació i amb quina clau d'encriptació s'ha fet. Si no tenim aquest dos paràmetres correctes, la matriu de commutació que ara tindrem no serà la exactament inversa i després del procés de mescal les subbandes continuaran permutades i el senyal continuarà sent un senyal encriptat.

En el cas d'encriptació hipotètic que s'ha estat exemplificant anteriorment, la estructura del procés d'anàlisi del desencriptador pertinent seria el següent:

Anàlisi:

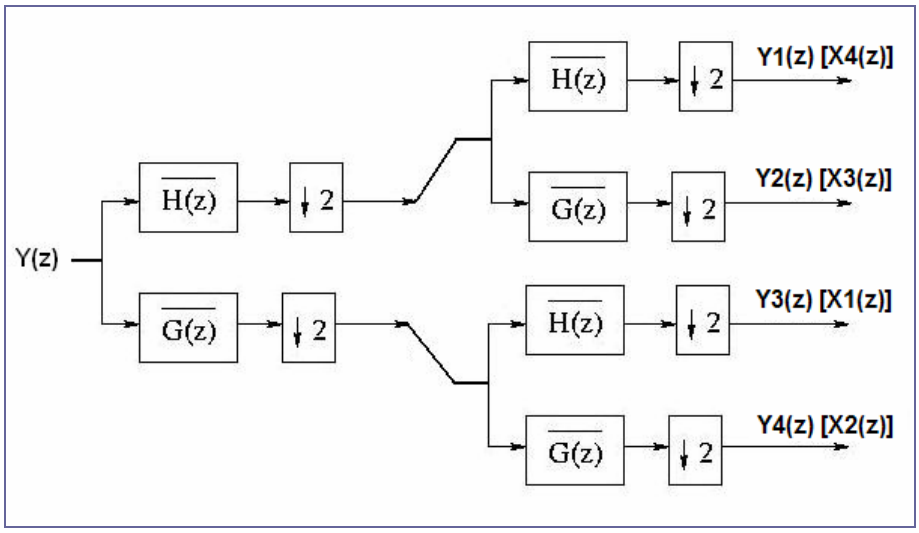

Figura3.9

En la *Figura3.9* es veu com es descompon el senyal encriptat en quatre subbandes de la mateixa manera com s'ha fet en la encriptació, passant cada bloc per dos filtres, un passa-alt i un passa-baix, obtenint-ne així dues subbandes, i repetint aquest procés per cada subbanda. També es delma després de cada filtratge per reduir el nombre de mostres a la meitat.

Alhora de recol·locar les subbandes allà on interessa per recuparar el senyal original, tornem a dependre d'una matriu de commutació, que al seu moment depèn de dos paràmetres, la clau d'encriptació i el nombre de subbandes. Així, per obtenir una recol·locació satisfactòria de les subbandes ens caldrà la clau d'encriptació i el nombre de subbandes correctes, de tal manera que ara crearem una matriu idènticament inversa a la d'encriptació, la seva transposada. Si la clau o el nombre de subbandes no són els correctes, la matriu que ara es crearà serà una matriu que no serà pas la matriu transposada de la que s'ha utilitzat alhora d'encriptar, i per tant les permutacions en aquest punt ens tornarien a deixar un senyal encriptat, amb informació d'una subbanda freqüencial en una altra, continuant el senyal encriptat i inintel·ligible.

Mescla:

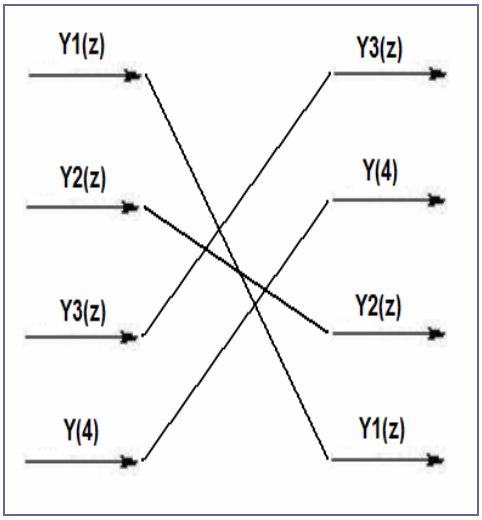

Figura3.10

Les recol·locacions que s'acaben de representar en la *Figura3.10* són les que el procés de desencriptació faria en el cas hipotètic que s'està plantejant al llarg d'aquesta explicació. Són les permutacions inverses a les que s'han realitzat en l'encriptació, donant per suposat que en aquest procés desencriptador tenim accés a les dades secretes d'encriptació, és a dir, al nombre de subbandes (n=4), i a la clau d'encriptació K.

Un cop fetes les reassignacions pertinents, com abans, només queda fer l'últim pas, el de recomposar el senyal a partir de les subbandes. El procés de Síntesi del desencriptador és exactament el mateix que el de l'encriptador: interpolem cada subbanda i la filtrem amb el filtre que li toqui. Després es sumen i es torna a fer el mateix procés fins a aconseguir tenir un sol senyal, el recompost.

Síntesi:

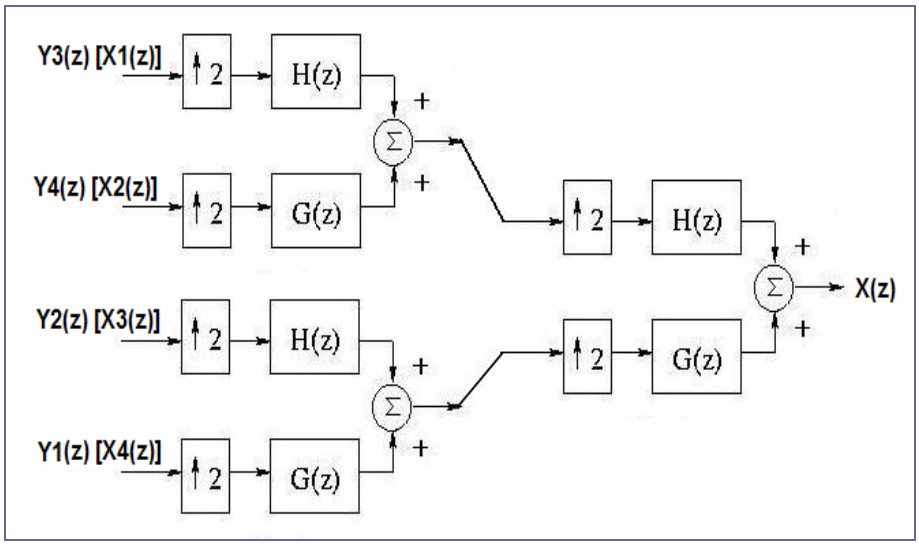

Figura3.11

Un cop finalitzat aquest procés de síntesi del desencriptador, si s'ha fet bé i amb les dades correctes, clau d'encriptació i nombre de subbandes, a la sortida s'obté el senyal original.

Aquest senyal original recompost pot patir certes alteracions, sobretot per l'efecte de filtratge, ja que no utilitza filtres ideals, si no *Chebishev*, que fa que cada vegada que filtrem alguna part del senyal, aquesta perdi una mica d'informació en els extrems. També influeix substancialment el nombre de subbandes en que es treballi, ja que evidentment, quantes més subbandes tinguem, un major nombre de vegades haurem utilitzat els filtres, i per tant més pèrdua de senyal tindrem.

3 3 Descomposició del senyal d'àudio en subbandes freqüencials

Com ja s'ha comentat aquest sistema encriptador/desencriptador es basa en la descomposició del senyal original de veu o àudio en subbandes freqüencials i la seva posterior permutació.

Per poder aconseguir aquesta separació espectral del senyal és necessari filtrar el senyal amb uns filtres que ens separin la informació en dos, és a dir, filtrar el mateix senyal amb un filtre passa-alt i una altre passa-baix, de manera que així tinguem una meitat de l'espectre per un costat i l'altra meitat per un altre. Les altes i les baixes freqüències per separat.

3 3 1 Filtratge

El filtratge serà necessari, com ja s'ha dit, per a poder fer aquesta descomposició del senyal d'àudio en altes i baixes freqüències.

Concretament utilitzarem dos filtres del tipus '*Chebyshev*'.

Són un tipus de filtres especials per a la separació de bandes de freqüència dels senyals, i una de les seves principals característiques és la seva velocitat de caiguda ('*roll-off*'), deguda a la implementació recursiva enlloc de convolutiva. Aquesta caiguda més vertical ens permet una major nitidesa alhora de filtrar. Són filtres dissenyats en base a una tècnica matemàtica anomenada '*z-transform*', o '*transformada z*' en català.

Es pot considerar que la resposta dels filtres Chebyshev és una estratègia matemàtica que intenta aconseguir una caiguda millor en detriment de permetre una major ondulació en la resposta freqüencial. Així, tots els filtres, tant analògics com digitals, que utilitzen aquesta estratègia són anomenats '*filtres Chebyshev*'. El nom prové de la utilització en aquests filtres dels '*polinomis de Chebyshev*', desenvolupats per el matemàtic rus Pafnuti Chebyshev (1821-1894). Al llarg de la història aquest nom ha estat traduït del rus de diferents maneres, i el podem trobar escrit com *Chebyshev*, *Tschebyscheff*, *Tchebysheff* o *Tchebichef*, per exemple.

Com ja hem introduït, la característica principal d'aquest tipus de filtres és que amb els càlculs matemàtics pertinents s'aconsegueix una caiguda més vertical de la resposta freqüencial del filtre en el punt de canvi de fase, però gràcies a que es permet un augment de les ondulacions en la zona de filtratge ('*ripple*').

Resposta en freqüència dels filtres Chebyshev:

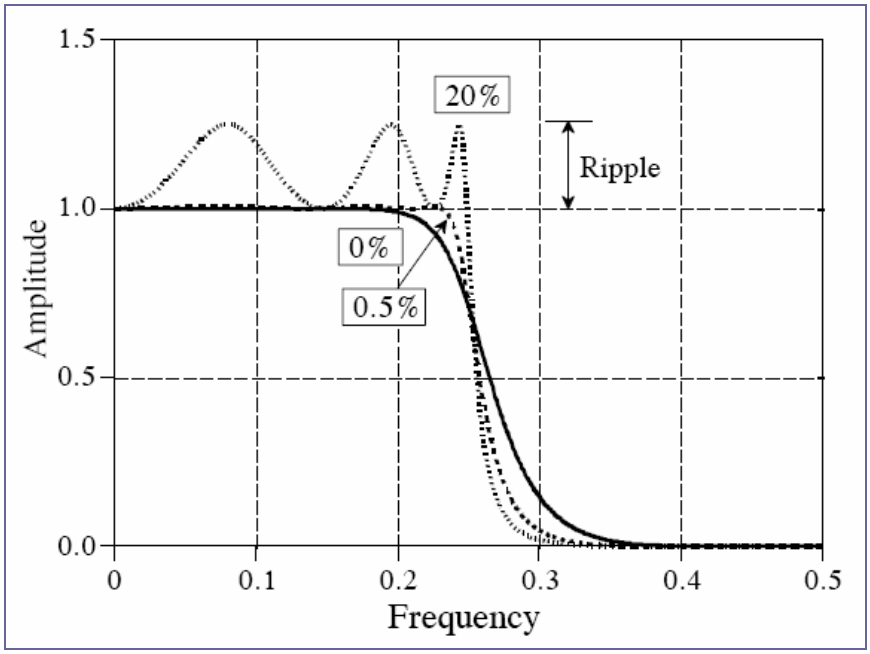

Figura3.12

En la *Figura3.12* podem comprovar que en la representació d'aquest filtre passa-baix del tipus *Chebyshev1*, observant les seves tres possibles respostes, si volem una caiguda més vertical (millor per filtrar) hem de permetre una ondulació major (pitjor). Així, la resposta dels filtres Chebyshev és una de les millors opcions de relació entre un fenomen i l'altre. En el cas en que tinguem una ondulació (*ripple*) del 0% ens trobarem aleshores davant un '*filtre de pla màxim*' o '*filtre de Butterworth*'. Anomenat així en honor a S. Butterworth, un enginyer anglès que al 1930 va descriure aquesta resposta.

Hi ha dops tipus de filtres Chebyshev, anomenats *Chebyshev1* i *Chebyshev2*. Es diferencien en que l'un, el primer tipus, permet aquestes ondulacions en la banda de pas ('*passband*') del filtre, i en canvi el segon tipus permet aquestes ondulacions en la banda d'atenuació ('*stopband*').

El nostre encriptador/desencriptador de veu l'hem implementat utilitzant dos filtres, l'un passa-baix i l'altre passa-alt, del tipus *Chebyshev2*, tenint així les ondulacions en la banda d'atenuació, i situant aquesta molt per sota del nivell del senyal per evitar possibles influències i imperfeccions en el filtratge.

El primer filtre, el passa-baix, l'hem implementat amb les especificacions següents: filtre de ordre 20 (amb ordre 10 ja faríem, però així ens assegurem un bon filtre), amb una atenuació de 80 dB (pot ser elevada, però també així assegurem un bon funcionament), i punt d'inflexió a 0'525 (el primer punt en que arribem als -80 dB d'atenuació).

Resposta en freqüència del filtre Chebyshev2 passa-baix creat:

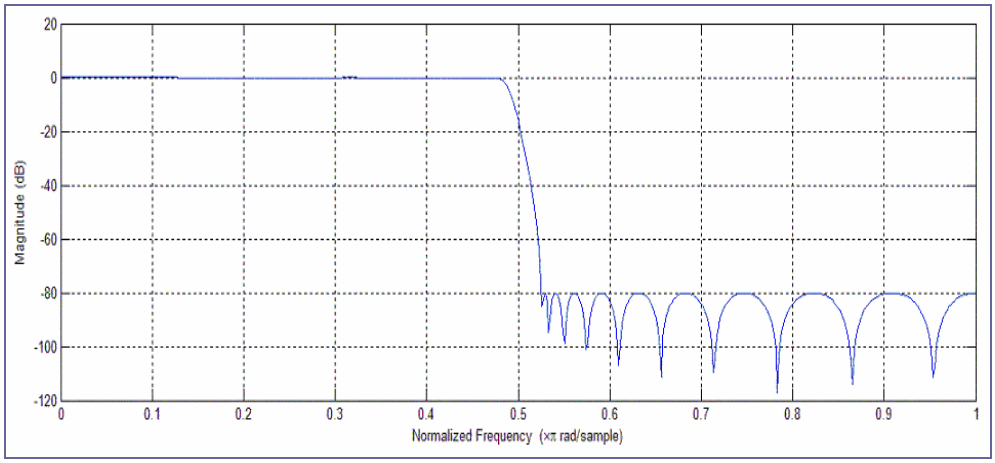

Figura3.13

En la resposta freqüencial representada en la *Figura3.13* veiem com es tracta d'un filtre Chebyshev2, ja que té les ondulacions en la banda d'atenuació. Banda que veiem que està als -80 dB respecte el senyal (0 dB), i que comença en el punt 0'525 del rang de freqüències. Això és la meitat de l'espectre del senyal, per poder separar-lo en dues meitats, les altes obtingudes amb aquest filtratge i les baixes freqüències que en traurem al filtrar per l'altre filtre.

El segon filtre creat, el pass-alt, segueix les mateixes especificacions que el primer, però variant una mica el punt d'inflexió. Aquest cop el situarem a 0'475, ja que ara serà el punt en que deixarà la zona d'atenuació per començar el flanc de pujada fins a la banda de pas.

Resposta en freqüència del filtre Chebyshev2 passa-baix:

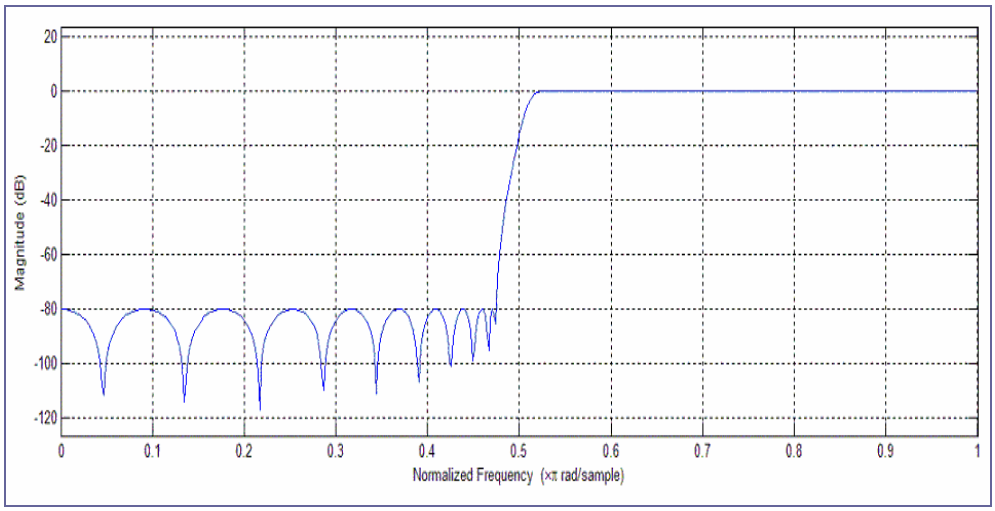

Figura3.14

Així, la combinació dels dos filtres ens agafa gairebé tot l'ample de banda. Trobem però a la part central de l'espectre, i degut a que és el punt de coincidència dels dos flancs, el de pujada per el filtre passa-alt i el de baixada per el passa-baix, una zona en que tindrem una mica de pèrdues de senyal. Aquestes pèrdues són poques, però depenent del nombre de vegades que es filtri el senyal ens afectaran més o menys, depenent el nombre de filtratges del nombre de subbandes en que estiguem treballant.

Aquesta combinació podem apreciar-la representant les dues respostes freqüencials, la de cada filtre, en un mateix gràfic, i veurem com en la zona central de l'espectre, on es creuen els dos flancs, la pèrdua de senyal hi serà present.

Però en canvi també podrem apreciar com al llarg de l'espectre tindrà una resposta molt plana, el que proporcionarà un bon filtratge, sense gaire modificacions.

Resposta freqüencial dels filtres Chebyshev2 passa-alt i passa-baix:

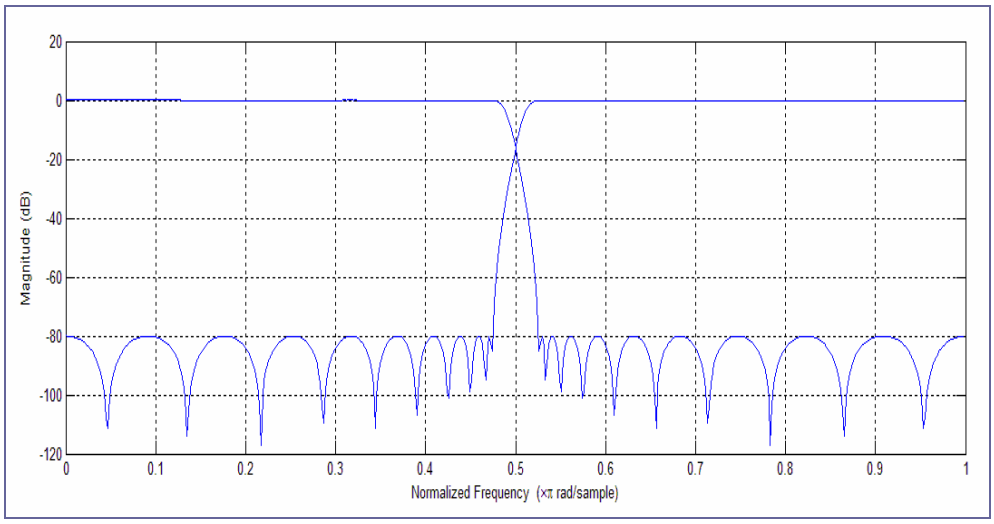

Figura3.15

La determinació dels dos punts d'inflexió, s'ha fet a partir de vàries proves on s'ha vist que si els situàvem els dos a 0'5, teníem una banda central amb molta pèrdua, i si els situàvem de manera que les dues bandes de pas coincidissin en els seus respectius extrems, just en aquest centre trobàvem un augment del senyal degut a la coincidència dels dos flancs, que quedaven sobreposats. La seva suma ens creava aquest augment del senyal en el rang central de freqüències.

Finalment, amb les especificacions fetes per als dos filtres, n'hem obtingut una molt bona resposta. Inevitablement tenim una mica de pèrdua en aquest rang de freqüències centrals, però és el menor possible.

3 3 2 Delmació i interpolació

La delmació i la interpolació són dos processos inversament iguals. El primer consisteix en eliminar d'un senyal discret, digitalitzat, una mostra de cada '*N*', sent '*N*' el nombre de la delmació. L'altre, la interpolació, fa el procés invers, ens afegeix una mostra cada '*N*' mostres, sent també '*N*' el nombre de

la interpolació. Aquestes dues operacions disminueixen o augmenten respectivament el nombre de mostres d'un senyal discret, augmentant o disminuint així la velocitat de mostreig del senyal continu que aquest discret representa. Recordem que qualsevol senyal digital d'àudio, per tant discret (*x[n]*), prové sempre d'un senyal analògic temporal (*x(t)*), o si més no, ho acaba sent. Per tant, no deixa de ser una seqüència numèrica que en algun moment ha estat i/o serà un senyal físic al llarg del temps, acabant sent-ne només una representació d'aquest úlltim.

En el procés d'encriptació/desencriptació són necessàries la delmació i la interpolació per poder anar filtrant iterativament. Això és perquè cal que en cada etapa, cada vegada que filtrem, reduïm també el nombre de mostres a la meitat, obtenint al final de l'execució de totes les etapes el tamany desitjat per a totes les subbandes, i després en el procés de recomposició cal doblar el nombre de mostres cada vegada que filtrem dues subbandes i les sumem, per la mateixa raó, hem d'anar doblant el nombre de mostres etapa a etapa fins a estar al nivell de les dimensions del senyal a recompondre, iguals a les del senyal original.

3 3 2 1 Delmació

La delmació és un procés que s'utilitza en casos en que tenim massa mostres en un senyal discret, i volem reduir-ne la freqüència de mostreig.

Sempre es delma per un nombre enter, el que permet jugar amb el tamany del senyar que es vol delmar i la seva freqüència de mostreig d'una manera molt simple, parlant sempre de múltiples simples.

Així, quan es parlar de delmar per *N* un senyal, el que es vol dir és que dividirem el tamany del senyal per *N*, així com la velocitat de mostreig del senyal pla subjacent.

Fixant-nos en el projecte, en els processos d'anàlisi que trobem tant en encriptació com en desencriptació, la delmació que durem a terme és una delmació per dos. Això és degut a la estructura d'arbre del procés. Recordem que començarem dividint el senyal en dues subbandes, després si cal en quatre, o en vuit o definitivament en setze. Així, en cada etapa del procés el que es fa és agafar un senyal o subbanda i dividir-lo en dues meitats. Per tant, cada vegada que dividim en dues meitats, comporta també la disminució a la meitat del nombre de mostres. Però el filtratge sigui el passa-alt o el passa-baix, només ens separa l'espectre de la banda filtrada en dos, de tal manera que les mostres pertinents a la banda de pas es mantenen igual, i les mostres de la banda d'atenuació pateixen una reducció considerable de l'amplitud, però en cap cas s'esborren, per tant sempre s'acaba mantenint el nombre de mostres.

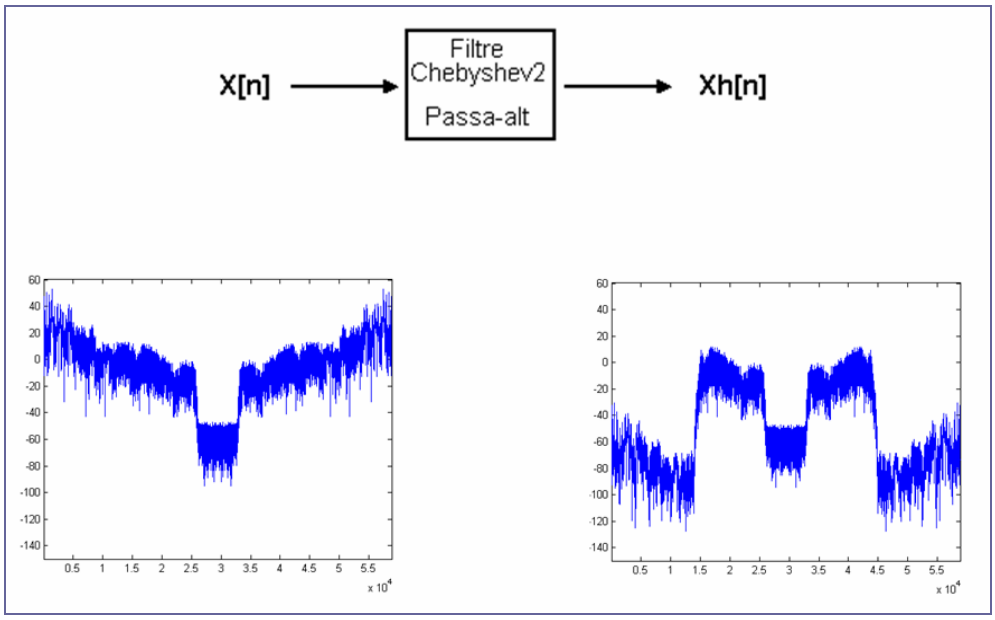

Figura3-16

Així doncs, un cop filtrat caldrà també reduir el nombre de mostres a la meitat, i per tant, delmar per dos.

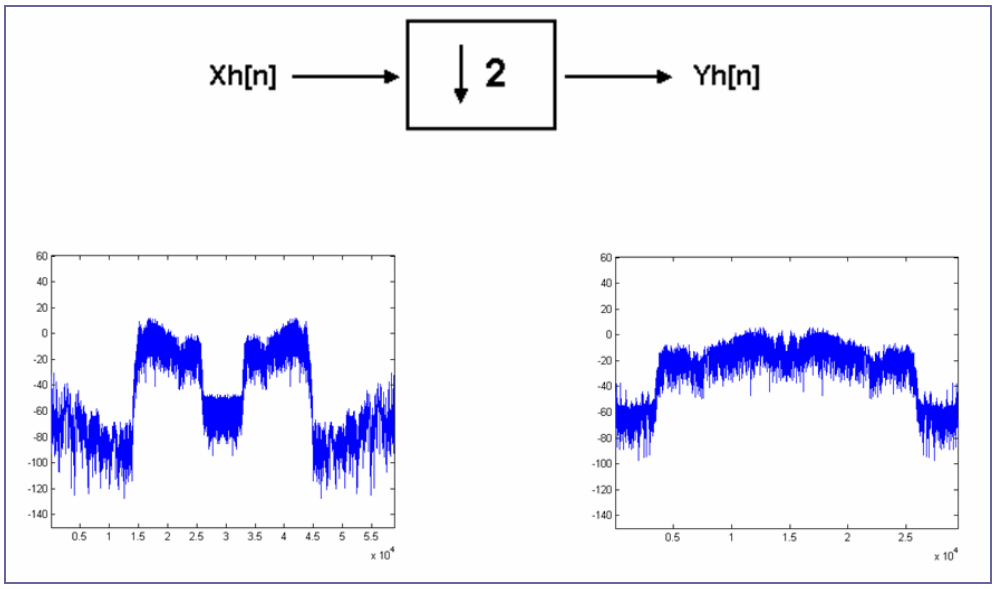

Figura3.17

Un cop feta la delmació per dos podem apreciar que el senyal resultant té la meitat de mostres (~ 3 x 10^4) que el senyal acabat de filtrar (~ 6 x 10^4). També podem veure que gràcies a la delmació, ara només tenim la part de l'espectre amb informació, la meitat que ens interessava i que per això hem filtrat.

Això és degut a que l'espectre, al delmar-se el senyal, es resitua en tota la banda espectral. Això vol dir que les dues parts que el conformen, recordem que l'espectre d'un senyal ens representa la seva distribució d'informació al llarg de tot el rang de freqüències, però dues vegades, com amb un mirall al mig. Així, la primera meitat s'expandirà al llarg de tot l'espectre, i la segona també, però inversament.

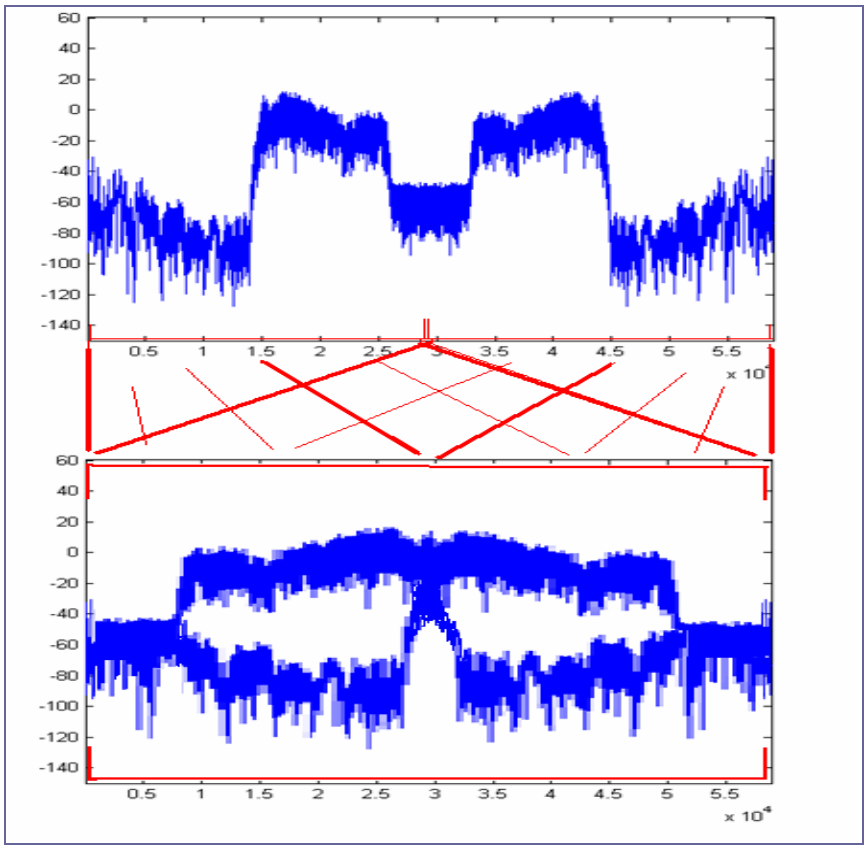

Figura3.18

Finalment, la delmació hem quedat que reduïa el tamany del senyal, així, ens quedarem amb el mateix espectre que un cop feta aquesta redistribució, però amb la meitat de mostres.

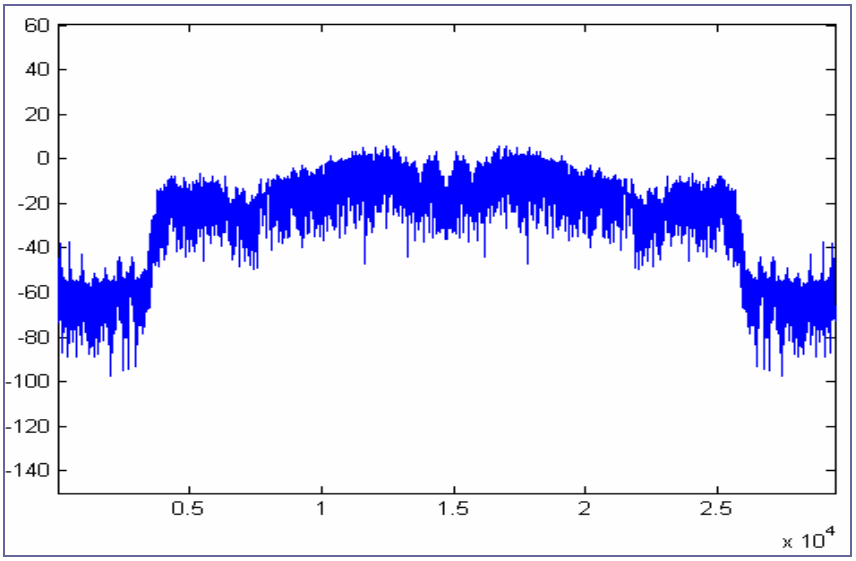

Figura3.19

L'exemple resultarà contenir informació de les altes freqüències, i senyal nul de les baixes, la seva distribució espectral se simplificarà a només la informació d'altes freqüències, ja que la diferència d'amplitud entre una informació i l'altra, fa que la segona no influeixi per res.

3 3 2 2 Interpolació

És el procés invers a la delmació, i comparteix amb ella les mateixes finalitats. És una eina que ens permet augmentar la velocitat de mostreig del senyal pla, així com recuperar predictivament un senyal delmat al seu tamany anterior, o directament augmentar-lo de tamany però amb una petita predicció que ens evita errors estrepitosos.

La interpolació consisteix en intercalar noves mostres entre les mostres del senyal que es pretén interpolar. Com la delmació, té un nombre que ens diu quina interpolació volem fer, i és aquest nombre el que ens marca quantes noves mostres intercalarem. Concretament, una interpolació per '**n**' vol dir que entre una mostra del senyal a interpolar i la següent, hi afegirem '*n-1*' mostres noves, situades en la fracció temporal corresponent per mantenir la continuïtat temporal del senyal.

Espectralment parlant la interpolació comporta una compressió de l'espectre freqüencial del senyal interpolat, en canvi temporalment, el senyal es veu expandit.

En el nostre procés necessitarem interpolar tantes vegades com s'hagi delmat, ja que l'utilitzarem per recuperar tant el tamany com la velocitat de mostreig que teníem inicialment, abans del procés d'anàlisi on l'hem delmat.

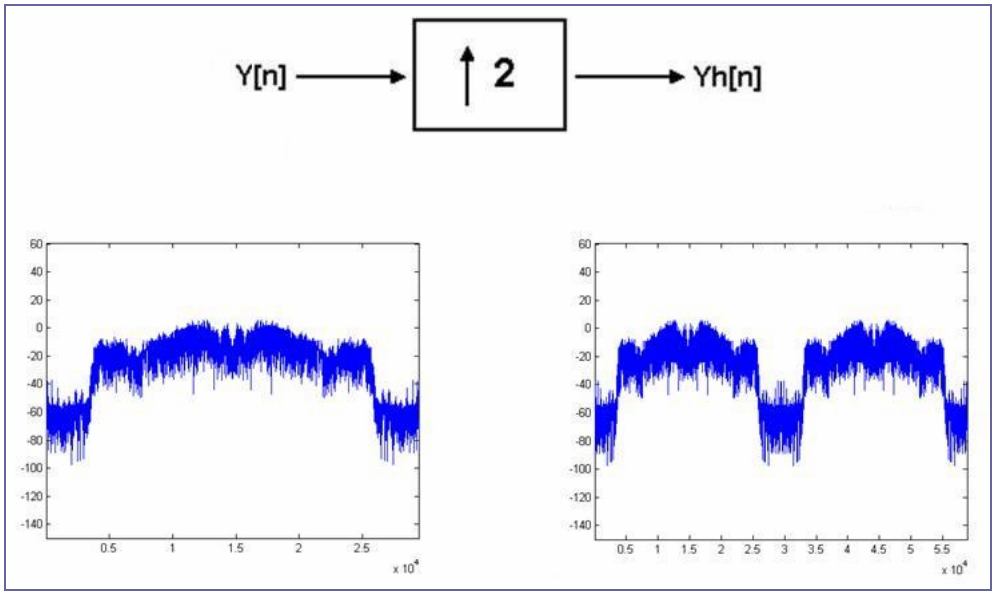

Figura3.20

En la *Figura3.20* podem apreciar els efectes d'interpolar un senyal. Al interpolar per 2 ara tenim el doble de mostres, i l'espectre queda de tal manera que s'ha doblat, com si fos un mirall.

En el nostre procés, un cop interpolada una subbanda, la filtrem per el filtre corresponent.

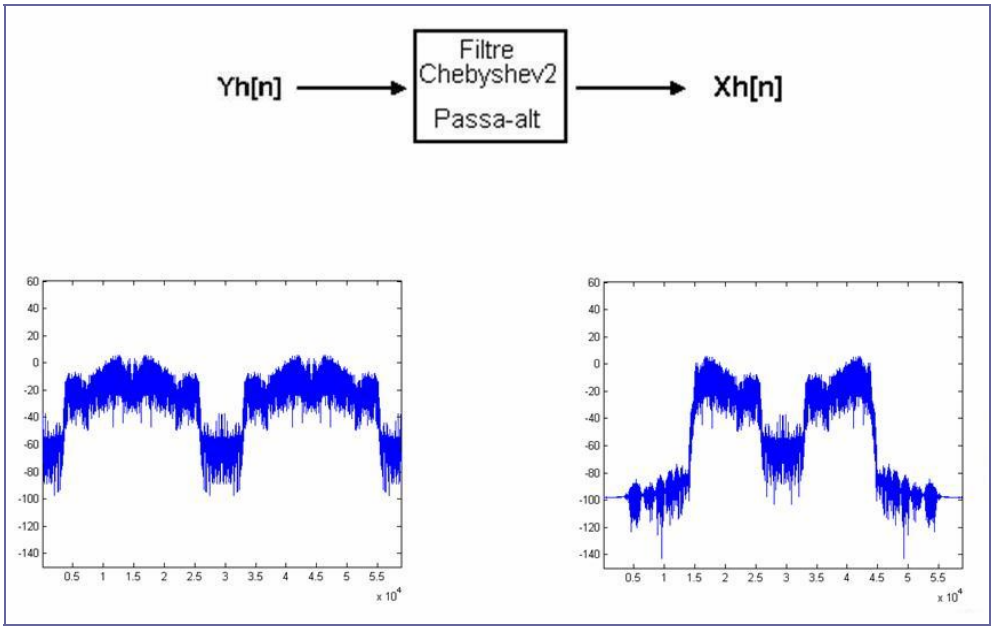

Figura3.21

I n'obtenim una subbanda de les que abans hem obtingut al filtrar una banda per dos filtres diferents. Per tant, ara només ens cal sumar aquesta subbanda a l'altre que obtindrem interpolant i filtrant per l'altre filtre. Si observem la *Figura3.11* de l'apartat **3 2 2**, que ens representa el procés de síntesi, es pot comprendre millor.

En l'exemple que s'ha vist al llarg d'aquesta explicació, arrancant des de la delmació, teníem una banda, l'hem filtrat passa-alt i delmat, i per recuperar-la, ara estem interpolant i filtrant de nou passa-alt. Un cop arribats a aquest punt, ens falta l'altra meitat de la informació de la banda inicial, ja que aquest és el procés passa-alt, però n'hi ha un de passa-baix paral·lel que està treballant amb la informació de greus. Per tant, ara per poder recuperar la banda inicial cal agafar la subbanda passa-baix que ha avançat en paral·lel a aquesta passa-alt, interpolar-la i filtrar-la passa-baix, i la subbanda resultant és aquesta informació de baixos que cal complementar amb la d'aguts que tenim per poder recuperar la banda inicial. Les sumem linealment i ja la tenim.

3 3 3 Influència de la freqüència de mostreig en el nombre de subbandes

Alhora d'escollir el nombre de subbandes en que es vol descompondre el senyal s'ha de tenir en compte la freqüència de mostreig.

La freqüència de mostreig és relativa a l'ample de banda, recordem que per complir el teorema de Nyquist i evitar així l'efecte d'*aliasing*, cal que sigui com a mínim del doble de l'ample de banda del senyal, i per tant, si resulta que el senyal d'entrada té una freqüència de mostreig petita, significarà que l'ample de banda també ho serà, i així fer una descomposició del senyal per a moltes subbandes serà innecessari. Sobretot perquè si tenim un ample de banda reduït vol dir que tenim la majoria de la informació útil del senyal concentrada en aquest ample, i per tant, si descomponem aquesta banda en poques

subbandes i les intercanviem de posicions en l'espectre, ja estem alterant gran part de la informació, modificant així notablement la intel·ligibilitat del missatge.

4 Implementació en Matlab de l'encriptador/ desencriptador de veu

4 1 Programa Principal

Com el seu nom indica, la funció que ens executa el programa principal té la funció bàsica d'ajuntar tots els components que per separat faran la feina desitjada, però que en algun lloc s'han d'executar en l'ordre correcte.

Per l'encriptador i el desencriptador les funcions del programa principal són gairebé idèntiques. La primera amb el nom de '*Encriptador_ok*' i la segona amb el nom de '*DesEncritpador_ok*'. Les dues reben com a paràmetres d'entrada el senyal que es pretén xifrar/desxifrar, la freqüència de mostreig d'aquest senyal, el nombre de subbandes en que farem la descomposició i la clau d'encriptació aleatòria, aquests dos últims paràmetres introduïts per l'usuari a partir de la interfície gràfica. I com a sortida, les dues funcions ens retornaran el senyal encriptat o el senyal original recuperat, depenent de quina estiguem parlant, si l'encriptador o el desencriptador respectivament. En ambdós casos, les dues funcions bàsicament faran tres crides, als tres processos anteriorment exposats, l'anàlisi, la mescla i la síntesi, seguint l'ordre adequat. Només es diferenciaran en un d'aquests tres processos, el de mescla. Així, l'anàlisi i la síntesi, sigui alhora d'encriptar o de desencriptar són processos idèntics, en canvi, la mescla no serà la mateixa quan parlem d'encriptar o de desencriptar.

Però al principi dels dos programes principals, abans de fer la crida als respectius processos d'anàlisi ens hem d'assegurar que el senyal d'entrada sigui en '*mono*', ja que les funcions s'han implementat per a senyals d'aquest tipus, no pas en estèreo. Així doncs, tenim una condició inicial que ens comprova si el senyal és '*mono*' o '*estèreo*', i depenent de com sigui, es fa una

suma dels dos canals (esquerra i dret pel cas de '*estèreo*'), o es treballa amb el mateix senyal d'entrada (si parlem d'un senyal '*mono*' de bon començament).

S'ha implementat així:

```
 9
10    % Si ens trobem amb un senyal d'àudio STEREO l'hem de passar a MONO
11 -  [j,i]=size(x);
12 -  if (i==2)% --> Si és STEREO
13        % separem els dos canals
14 -      xL=x(1:j,1);
15 -      xR=x(1:j,2);
16        % sumem els dos canals, pero amb la meitat de potència del senyal
17 -      x_mono = (xL/2) + (xR/2);
18 -  else % --> Si és MONO
19 -      x_mono = x;
20 -  end
21
```

Un cop ja tenim el senyal en *mono*, en un sol canal, el que fem és la crida ordenada de les altres funcions.

Comencem fent l'anàlisi, fent la crida a la funció '*BandesVaries*', la que ens descompondrà el senyal d'entrada en el nombre de subbandes desitjat.

```
22    % Descomposem el senyal en el nombre de bandes dessitjat
23 -  [sb,E] = BandesVaries(x_mono,fm,n);
```

Un cop descomposada la senyal d'entrada, i emmagatzemada cada subbanda en un dels vectors de la matriu '*sb*', és el moment del procés de mescla. És en aquest punt que l'encriptador farà la crida a una funció i el desencriptador a una altra.

En el cas de l''*Encriptador_ok*', per aquest procés de mescla s'ha creat una funció anomenada '*barreja_Encript*', i per el cas del '*DesEncriptador_ok*' la funció creada s'anomena '*barreja_Desencript*'.

Encriptador:

```
25    % Fem la reassignació de les subbandes
26 -  [sb_out] = barreja_Encript(sb,g);
```

Desencriptador:

```
25    % Fem la reassignació de les Subbandes
26 -  [sb_out] = barreja_Desencript(sb,g);
```

En ambdós casos, els paràmetres d'entrada o sortida són els mateixos, però reben un tracte diferent dintre la funció. Les dues funcions depenen de la matriu '*sb*' on hi tenim cada subbanda, i de la clau d'encriptació aleatòria, en aquest cas '*g*'. A la sortida ens retornen una matriu de les mateixes dimensions que '*sb*', però que conté les subbandes en un ordre diferent, i que s'anomena '*sb_out'*. En el cas de la encriptació, '*sb*' conté les subbandes en l'ordre del senyal original i '*sb_out*' conté les subbandes permutades. En el desencdriptador les característiques de '*sb*' i '*sb_out*' són les inverses, el primer amb les subbandes desordenades, i el segon, si la clau d'encriptació i el nombre de subbandes han estat els correctes, amb les subbandes ordenades igual que el senyal original.

Un cop fet el procés de mescla, entrem en el procés de síntesi. Ara tant l'encriptador com el desencriptador tornen a coincidir, fent la crida a la mateix funció. Funció '*Reconstruct_BandesVaries*', la qual ens retorna un senyal únic '*y*', representat amb un sol vector, compost a partir de la matriu '*sb_out*' que ens contenia les subbandes del senyal per separat.

```
28    % Reajuntem les subbandes per obtenir el Senyal Encriptat
29 -  [y] = Reconstruct_BandesVaries(sb_out,fm);
```

En el cas de la encriptació, '*sb_out*' contindrà les subbandes desordenades, i per tant '*y*' serà el senyal encriptat, i en canvi en el cas del desencriptador, '*sb_out*' contindrà les subbandes reordenades, i per tant '*y*' serà la recomposició del senyal original.

Aquí acaba la feina del programa principal. A partir d'aquí ja ens hem d'endinsar en els tres processos acabats d'exposar.

4 2 Separació del senyal d'àudio en subbandes

Com acabem de veure, tant l'encriptador com el desencriptador utilitzen la mateixa funció per dur a terme el procés d'anàlisi, la descomposició del senyal en subbandes. Aquesta funció comuna hem vist que s'anomena '*BandesVaries*', i la seva capçalera és:

```
1   % Funció 'BandesVaries'
2
3   function [sb,E] = BandesVaries(x,fm,n)
4                       %% sb => Matriu amb el senyal separat en subbanda
5                       %% E => Comentari que ens apareixerà per la pantalla de
6                       %%      comandes del Matlab
7                       %% x => Senyal Original
8                       %% fm => Freqüència de Mostreig
9                       %% n => Nº de Subbandes
10
```

Es pot veure quins són els paràmetres d'entrada i sortida de la funció. '*x*' en aquest cas és el senyal original ja que ho hem extret del procés d'encriptació, però en el procés de desencriptació serà el senyal encriptat. En efecte, més que res és el senyal d'entrada que a la sortida tindrem descomposat en '*n*' subbandes i emmagatzemat en la matriu '*sb*'. I '*fm*' és la

freqüència de mostreig d'aquest senyal d'entrada, sigui l'original o l'encriptat. Finalment tenim '*E*' que és una variable que ens permetrà mostrar un missatge per la pantalla de comandes del Matlab, però un cop ficat tot el sistema encriptador/desencriptador dins de la interfície gràfica que s'ha creat, aquest missatge no el podrem veure.

La funció comença verificant el nombre de subbandes escollit per l'usuari. El programa creat només accepta fer descomposició del senyal en 2, 4, 8 o 16 subbandes freqüencials, i per tant, per poder començar a treballar cal que es comprovi que el nombre de subbandes escollit per l'usuari i que ens ve introduït per la variable '*n*' sigui un d'aquests quatre valors.

```
 11
 12    % Només farem càlculs si hem escollit separar el Senyal Original en 2, 4,
 13    % 8, o 16 Subbandes
 14 -  if ((n==2) | (n==4) | (n==8)| (n==16))
 15
---- -
---- -
---- -              (COS DE LA FUNCIÓ)
---- -
---- -
---- -
201
202 -  else% --> En el cas d'haver escollit un nº de Subbandes diferent a
203        %      2, 4, 8 o 16, mostrem un ERROR per pantalla i triem una matriu
204        %      buida
205
206 -      E= 'ERROR! Nombre de subbandes INCORRECTE, ha d"escollir entre 2,4,8 o 16 subbandes'
207 -      sb=0;
208
209 -  end
210
```

En el cas de que el nombre de subbandes no sigui un dels que el programa permet, podem apreciar com mitjançant la variable '*E*' es mostrarà per pantalla un missatge que avisa de l'error i dóna les opcions correctes per poder executar bé l'encriptador/desencriptador. En conseqüència d'aquest error, la matriu de sortida '*sb*' sortirà buida.

Si per el contrari, el nombre de subbandes escollit és un dels correctes, entrarem dins del cos principal de la funció, on farem la descomposició del senyal '*x*' d'entrada en '*n*' subbandes.

Començarem per decidir el nombre de subbandes en que farem la descomposició, depenent no només del nombre escollit per l'usuari, si no fixant-nos en la freqüència de mostreig del senyal introduït. Com ja s'ha comentat anteriorment, limitarem el nombre de subbandes en funció de la freqüència de mostreig.

Freqüència de mostreig	2 subband.	4 subband.	8 subband.	16 subband.
0 Hz --> 10000 Hz	SÍ	SÍ	NO	NO
10 KHz --> 30 KHz	SÍ	SÍ	SÍ	NO
> 30 KHz	SÍ	SÍ	SÍ	SÍ

Figura4.1

Així, per dur a terme aquestes limitacions, treballem amb una varibale auxiliar que anomenem '*nb*' i que serà el nombre que finalment utilitzarem com a nombre de subbandes. Un cop decidit el valor final de '*nb*', mostrarem un missatge per la pantalla de comandes del Matlab enunciant el nombre de subbandes amb que finalment el programa treballarà.

```
24        %% Depenent de la freq de mostreig limitarem el nombre de subbandes, ja
25        %% que per a 'fm' baixes és absurd descompondre en moltes subbandes
26 -      if ((fm<10000)&(n>4))
27 -          nb = 4;
28 -      elseif ((fm>10000)&(fm<30000)&(n>8))
29 -          nb = 8;
30 -      elseif ((fm>30000)&(n>16))
31 -          nb = 16;
32 -      else
33                        %% si no s'ha cumplert cap de les possibilitats
34 -          nb = n;       %% anteriors, farem els càlculs amb el nombre de
35                        %% subbandes proposat per l'usuari
36 -      end
37
38        % Treurem un missatge per pantalla:
39        % 'OK, farem el procés amb -nb- subbandes. Procés en curs...'
40 -      a = 'OK, farem el procés amb -';
41 -      e = '- subbandes. Procés en curs...';
42 -      i = nb;
43
44 -      E = strcat(a,num2str(i),e)
45
```

El missatge que traurem per pantalla serà mitjançant la funció *'strcat'*, una funció predefinida de Matlab que ens permet ajuntar cadenes de caràcters. Així, i mitjançant una altra funció de Matlab que ens permet passar una variable de nombre a cadena de caràcters (*'num2str'*), ho ajuntarem tot i per pantalla ens sortirà el missatge que ens dirà quin ha estat finalment el nombre de subbandes amb que farem tot el procés.

A partir d'ara ja comencem a entrar en matèria.

Inicialment, abans de fer res, cal que creem els dos filtres que utilitzarem al llarg del procés d'anàlisi. Recordem que seran dos filtres iguals, però invertits, i del tipus *Chebyshev2*. L'un passa-baix i l'altre passa-alt.

```
15
16        % Creem 2 filtres chebytchez, un passa-alt i un passa-baix
17 -      [Bl,Al]=cheby2(20,80,0.525);% --> Filtre passa-baix
18 -      [Bh,Ah]=cheby2(20,80,0.475,'high');% --> Filtre passa-alt
19
```

La funció '*cheby2*' és del Matlab mateix, i ens crea el filtre *Chebyshev2* que volem. Cal que li definim els paràmetres que abans hem comentat. Així, el 20 ens indica que és un filtre d'ordre 20, el 80 ens marca la caiguda en dB's de la zona d'atenuació respecte el senyal que s'està filtrant (nivell d'entrada és 0dB), i el punt del rang freqüencial en que canviarà de fase. El primer, que ens crea el filtre passa-baix, té aquest punt d'inflexió en 0'525, i el passa-alt el definim a 0'475, com ja hem comentat anteriorment, per evitar pèrdues d'informació o augments de nivell per culpa de la manca o la superposició de les respostes dels dos filtres.

La funció ens retorna *B* i *A*, que són els coeficients del numerador i del denominador respectivament de la funció de transferència del filtre que ens crea.

Un cop tenim els filtre ja podem començar el procés de filtratge.

Recordem que es segueix un procés amb estructura l'arbre. Per tant, comencem descomposant el senyal en dues subbandes. Si en volem més de dues, tornem a filtrar cada subbanda i n'obtenim quatre, i si resulta que també en volem més de quatre, tornem a repetir el procés i ara en tenim vuit. I en el cas de més de vuit, ho repetim de nou i n'obtenim setze.

Recordem que cada vegada que filtrem, tot seguit delmem, per no tenir accés de tamany.

```
46        % Comencem els càlculs
47        % Filtrem les dues primeres bandes
48 -      xl = filter (Bl,Al,x);
49 -      xh = filter (Bh,Ah,x);
50
51        % Delmem les dues primeres bandes
52 -      yl = xl(1:2:length(xl));
53 -      yh = xh(1:2:length(xh));
54
```

El filtratge el fem amb al funció de Matlab '*filter*', introduint-li com a valors d'entrada els coeficients que abans hem obtingut en la creació del filtre (*B* i *A*) i el senyal que es vol filtrar. En el cas de la descomposició d'un senyal en dues parts, en els dos filtratges introduïm el mateix senyal d'entrada, però l'un amb els coeficients del filtre passa-baix (*Bl* i *Al*) i l'altre amb els del passa-alt (*Bh* i *Ah*). I a la sortida en traiem la informació de baixes freqüències, de la meitat en avall, en una variable i les altes, de meitat en amunt, un una altra.

Per que el filtratge sigui coherent cal que ara cada subbanda tingui la meitat del tamany que el senyal d'entrada, així, cal que els delmem per dos. I matemàticament la delmació per dos consisteix en quedar-se amb la meitat de mostres del vector, així que n'emmagatzemarem una de cada dos en un vector auxiliar.

En el cas de que s'hagi decidit fer la descomposició en més de dues subbandes, entrem en un procés que ens va desglossant el senyal mica en mica, fins a tenir el nombre de subbandes desitjat.

En cada etapa seguim tres passos. El primer és filtrar, per obtenir el doble de subbandes, després delmar cada subbanda obtinguda, i per últim, i si ja estem en el nombre de subbandes desitjat, inicialitzem la matriu de sortida '*sb*' amb el tamany adequat i hi emmagatzemem cada subbanda en un dels vectors que componen la matriu. Si en canvi, després del segon pas resulta que encara no tenim el nombre de subbandes desitjat, tornarem a filtrar cada subbanda per doblar-ne el nombre, tornarem a delmar, etc... i així fins a arribar al nombre de subbandes desitjat, i per tant, fins a tenir creada la matriu '*sb*' de sortida amb el nombre de vectors igual al de subbandes i cadascun d'ells amb el mateix tamany que el de les subbandes finals. Així queda emmagatzemada cada subbanda del senyal d'entrada en un dels vector que componen la matriu '*sb*'. Quedaran les subbandes ordenades de baixes a altes freqüències, ordre important ja que alhora de la desencriptació haurem de tenir clar on tenim la informació, i on cal situar-la.

```
 55 -       if ( nb>3 )% --> Si volem fer-ho amb més de 3 Subbandes
 56             %% filtrem les 2 bandes per obtenir-ne 4
---  -             -filtratges-
 63             % Delmem les quatre bandes
---  -             -delmacions-
 69 -           if(nb==4)% --> Si hem escollit fer-ho per 4 Subbandes, ja podem
 70                      %     guardar les Subbandes que ja tenim
 71
 72             % Guardem les 4 Subbandes
 73 -           sb = zeros(length(yll),nb);% --> Inicialització de la matriu 'sb'
 74
 75 -           sb(1:length(yll),1) = yll;  %% Baixes freqs
 76 -           sb(1:length(yll),2) = ylh;
 77 -           sb(1:length(yll),3) = yhl;
 78 -           sb(1:length(yll),4) = yhh;  %% Altes freqs
 79
 80 -           else% --> Si teníem més de 3 Subbandes, pero no són 4, continuem
 81                 % Filtrem les 4 Subbandes per obtenir-ne 8
---  -               -filtratges-
 95                 % Delmem les 8 Subbandes
---  -               -delmacions-
106 -               if(nb==8)% --> Si hem escollit fer-ho per 8 Subbandes, ja
107                         %     podem guardar les Subbandes que ja tenim
---  -               -emmagatzements-
121 -               else% --> Si teníem més de 3 Subbandes,però no són 8,continuem
122                     % Filtrem les 8 Subbandes per obtenir-ne 16
---  -                   -filtratges-
149                     % Delmem les 16 Subbandes
---  -                   -delmacions-
167                     % Guardem les 16 subbandes
---  -                   -emmagatzements-
187                 end
188
189 -           end
190
191 -       else
192             % Aquí només entrarem en el cas de que nb<3, tenint 2 subbandes
193             % les emmagatzemem
194 -           sb = zeros(length(yl),nb);% --> Inicialització de 'sb'
195
196 -           sb(1:length(yl),1) = yl; %% Baixes freqs
197 -           sb(1:length(yl),2) = yh; %% Altes freqs
198 -       end
```

Amb el codi podem apreciar la estructura bàsica de la funció, sense entrar en tots els càlculs, ja que són idèntics als ja vistos, però cada vegada repetits més vegades.

Al final apreciem què és el que passa si el nombre de subbandes escollit és dos. Com ja havíem fet el primer filtratge al inici dels càlculs i per tant ja

teníem les dues primeres subbandes, només cal que inicialitzem el vector de sortida '*sb*' i que emmagatzemem les dues subbandes en els dos vector que conformen '*sb*'.

Aquí acaba la descomposició en subbandes. Ja tenim una matriu anomenada '*sb*' que conté el senyal d'entrada desfragmentat en el nombre de subbandes freqüencials que l'usuari ha decidit.

Ara, i passant pel programa principal, toca el procés de mescla. En un cas serà desordenar de les subbandes i en un altre tocarà reordenar-les, depenent de si estem encriptant o desencriptant.

4 3 Mescla de les subbandes

Entrem en el procés de mescla. Com ja hem comentat vàries vegades, el procés no serà el mateix si parlem de la mescla quan encriptem o quan desencriptem.

4 3 1 Mescla en encriptació

La mescla en encriptació la fem amb la funció '*barreja_Encript*'. És una funció que té una matriu d'entrada, concretament la '*sb*' obtinguda en la funció '*BandesVaries*' executada en el programa principal d'encriptació, i que per tant, conté el senyal original fragmentat en subbandes freqüencials i ordenades de baixes a altes freqüències. A la sortida, aquest funció ens retorna una matriu '*sb_out*' de les mateixes dimensions que la d'entrada, i amb la mateixa informació, però en un ordre diferent. L'ordre de sortida dels vector que

conformen '*sb*' ve determinat per una matriu de commutació creada a partir de la clau d'encriptació aleatòria introduïda per l'usuari.

```
3   function [sb_out] = barreja_Encript(sb_in,codi)
4               %% sb_out => vector de sortida, amb les bandes d freq. desordenades
5               %% sb_in => vector on tenim guardades les bandes d freq. del
6               %%              senyal original
7               %% codi => Codi d'Encriptació
```

Un cop dintre la funció, el primer que s'ha de fer és crear la matriu de sortida '*sb_out*'. Com és de les mateixes dimensions que la d'entrada, n'hem de prendre les d'aquest primera de referència.

```
9   % Obtenim les dimensions del vector d'entrada
10  [m,n] = size(sb_in);
11          %% m => Tamany de cada banda de freq del senyal de sortida
12          %% n => Nombre de bandes de frequències
13
14  % Inicialitzem el vector de sortida, que tindrà le dimensions del d'entrada
15  sb_out = zeros(m,n);
16
```

El funcionament de la funció consisteix en canviar els vectors d'ordre, de tal manera que quan fem la recomposició en el procés de síntesi, el senyal que obtinguem estigui encriptat, ja que les subbandes freqüencials han estat permutades i així la informació que tindrem a la sortida de cada subbanda no serà la corresponent.

En el cas de que estiguem treballant sobre dues subbandes, el que farem és simplement canviar els vectors de lloc. El vector que conté la informació d'altes freqüències el guardarem en el lloc del que conté les baixes, i viceversa.

```
16
17 -  if(n==2)
18        % Tenint només 2 Subbandes, no cal matriu de conmutació, simplement
19        % invertim els senyal
20
21 -      sb_out((1:m),1)= sb_in((1:m),2);
22 -      sb_out((1:m),2)= sb_in((1:m),1);
23
```

En canvi, en el cas de tenir més de dues subbandes, la permutació de les subbandes la farem a partir d'una matriu de commutació. Aquesta matriu de commutació es genera amb de la funció '*genmatriu*', que depèn de la clau d'encriptació i del nombre de subbandes que ha introduït l'usuari. Aquí és on recau el pes del sistema. La correcta obtenció d'aquests dos paràmetres és el que farà que es creï una matriu de commutació o una altra i per tant, que el sistema encriptador/desencriptador funcioni satisfactòriament.

```
23
24 -  else
25
26        % Obtenim la matriu de commutació que utilitzarem, depenent del nombre de
27        %   subbandes i del Codi d'Encriptació amb la funció 'genmatriu'
28 -      matriu = genmatriu(codi,n);
29
30        % Amb aquest bucle recol·loquem cada Subbanda on li toqui, seguint la
31        % matriu de conmutació ('matriu')
32 -      for N = 1:n
33
34 -          [j,i]= max(matriu(1:n,N));
35                % i = posició de la fila on hi ha l'únic 1, serà la banda per on
36                %     treurem cada banda
37
38 -          sb_out((1:m),i)= sb_in((1:m),N);
39
40 -      end
41
42 -  end
43
```

Com podem comprovar, un cop tenim la matriu de commutació, el que hem de fer és trobar on hem de col·locar cada subbanda. Aquest bucle '*for*' explora la matriu fila a fila, trobant en cada iteració la posició de l'únic '1' que

conté la fila explorada, posició que ens determina per a quin canal de sortida, és a dir vector, hem de treure el vector d'entrada a qui ens estem referint en cada repetició.

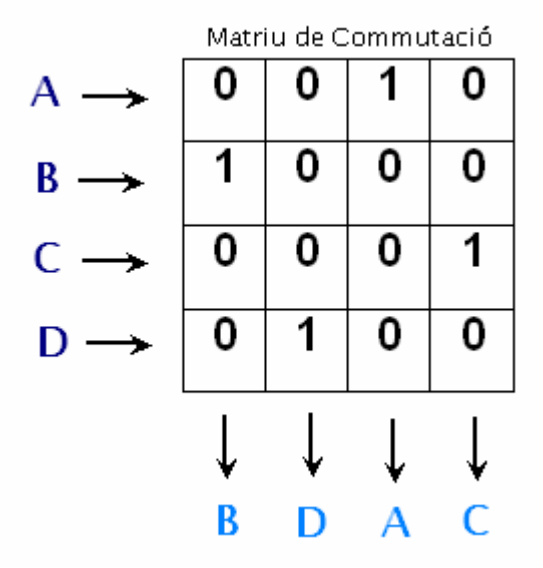

Figura4.2

En aquest cas podem observar que la '*A*' entra com a '*vector 1*', i la matriu de commutació ens diu que el que entri per la primera entrada, haurà de sortir per el '*vector 3*', ja que és en la tercera posició de la primera fila on es troba el bit '1'. I així per a tots els vectors d'entrada, fixant-nos per a cadascun en la fila de la matriu de commutació corresponent.

El bucle que comentàvem anteriorment fa aquesta funció, decidir cada vector d'entrada per on haurà de sortir, seguint la matriu de commutació com a guia.

4 3 2 Mescla en desencriptació

Ja s'ha comentat que el procés és gairebé el mateix, exceptuant el tracte que es dóna a la matriu de commutació.

Ara, la matriu d'entrada representa els senyal que he estat encriptat, separat per subbandes, així que si la descomposició s'ha fet en el nombre de subbandes correcte, és a dir, en el mateix nombre de subbandes amb que s'ha encriptat, en cada vector tindrem la informació d'una subbanda, però sense estar ordenats. La matriu de sortida, si el codi i el nombre de subbandes són els correctes, serà el senyal original recomposat i separat per subbandes. preparat per passar al procés de síntesi que ens el reagruparà i ens tornarà el senyal original.

```
3   function [sb_out] = barreja_Desencript(sb_in,codi)
4               %% sb_out => vector de sortida, amb les subbandes recol·locades
5               %%           on els toca per tornar a tenir el Senyal Original
6               %% sb_in => vector d'entrada amb les subbandes desordenades
7               %% codi => codi de Desencriptació
```

El procés comença exactament igual que en la mescla de la encriptació, inicialitzant la matriu de sortida amb les dimensions de la d'entrada i fent la permutació de les subbandes per el cas de treballar només amb dues.

És en el moment d'utilitzar la matriu de commutació que es diferència un procés d'un altre. Per el primer cas hem vist que a partir de la clau d'encriptació i del nombre de subbandes obteníem una matriu de commutació que ens permutava les subbandes, i ara el que estem intentant és invertir aquesta permutació, és a dir, que cada subbanda torni allà on li pertoca. Per tant, per poder fer això el que necessitem és una matriu de commutació que sigui exactament la matriu 'inversa' a la que hem utilitzat en encriptar. Aquest matriu 'inversa' és la matriu transposada de la primera. Així doncs, per poder tornar cada subbanda a la part de l'espectre de freqüències que li pertoca necessitem la matriu transposada a la que s'ha utilitzat en encriptar, i la millor manera d'aconseguir-la és obtenir aquesta primera matriu i calcular-ne la seva transposada. És per això que la clau aleatòria i el nombre de subbandes correctes són tant necessaris per desencriptar, ja que necessitem tornar a obtenir la mateixa matriu de commutació inicial, per després transposar-la i poder utilitzar aquesta última per tornar cada subbanda allà on li pertoca.

```
24
25 -  else
26
27        % Obtenim la matriu de commutació que utilitzarem, depenent del nombre
28        % de subbandes i del Codi de Desencriptació amb la funció 'genmatriu'
29 -      matriu = genmatriu(codi,n);
30 -      tras_matriu =  matriu';
31
32        % Amb aquest bucle recol·loquem cada Subbanda on li toqui, seguint la
33        % trasposada de la matriu de conmutació ('tras_matriu')
34 -      for N = 1:n
35
36 -          [j,i]= max(tras_matriu(1:n,N));
37              % i = posició de la fila on hi ha l'únic 1, serà la banda per on
38              %     treurem la banda actual
39
40 -          sb_out((1:m),i)= sb_in((1:m),N);
41
42 -      end
43
44 -  end
45
```

Podem apreciar com el funcionament és el mateix que en el cas d'encriptar, amb la diferència que ara treballarem amb una variable anomenada '*tras_matriu*', que correspon a la matriu de commutació trobada amb a funció '*genmatriu*' i transposada.

Així, si ens fixem en l'exemple de matriu de commutació que hem donat anteriorment (*Figura4.2*), i resulta que ens trobem en el moment de recol·locar les subbandes que aquella matriu ens ha permutat, ara, la matriu de commutació que necessiteríem és la seva transposada.

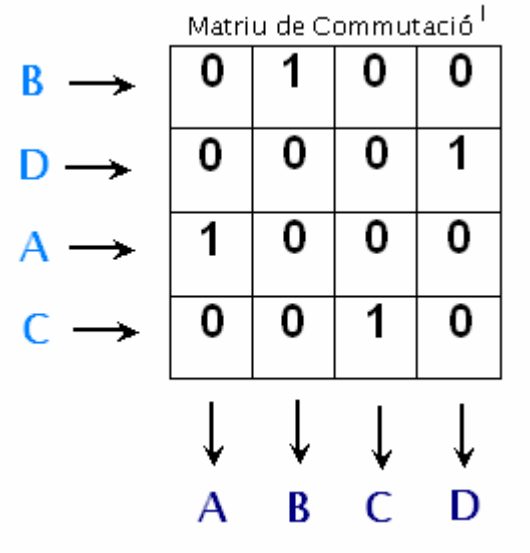

Figura4.3

Així doncs, veiem que utilitzant la transposada de la matriu que ens ha encriptat, podem recol·locar cada subbanda allà on li pertoca, recuperant així el senyal original.

4 3 3 Matrius de commutació

Per a fer les permutacions de les subbandes, ja hem comentat que es faran a partir d'unes matrius de commutació que crearem cada vegada, sempre en funció del nombre de subbandes i de la clau d'encriptació introduïts per l'usuari. Així, la funció '*genmatriu*' ens crearà matrius de commutació depenent d'aquests dos paràmetres d'entrada.

Una matriu de commutació és una matriu que ens definirà per on sortirà un valor que entra per una de les entrades. Així, si el que volem és canviar de posició *M* vectors (valors, etc...), tindrem *M* entrades i *M* sortides. Per tant, necessitem una matriu de commutació de dimensions *M*x*M*, és a dir, *M* vectors de M mostres cadascun, que ens equival a una matriu de *M* files per *M* columnes. I perquè sigui una matriu de commutació és condició indispensable que la matriu sigui tota de *zeros* ('0') exceptuant *M uns* ('1'). I cal que aquests

estiguin distribuïts de tal manera que només trobem un '1' per fila i un '1' per columna.

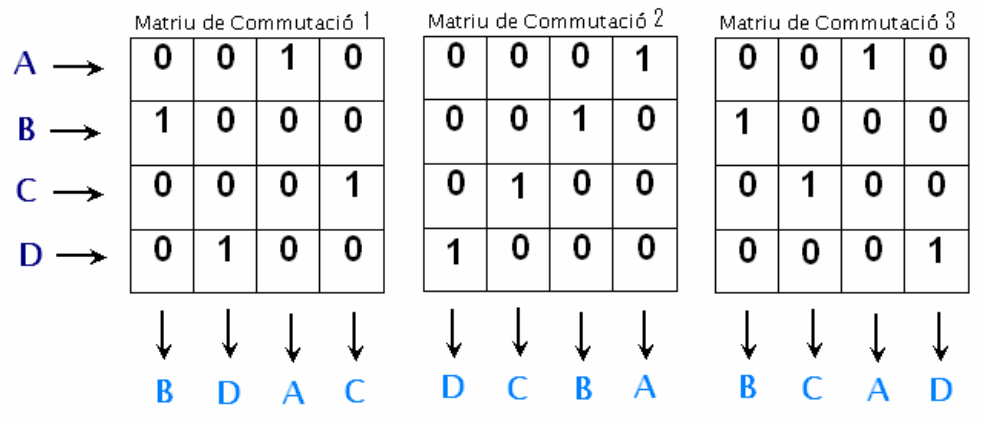

Figura4.4

En aquest exemple es veu com depenent de la distribució d'aquests *M uns* dintre la matriu de commutació, i sempre respectant que només n'hi hagi un per fila i un per columna, l'ordre de sortida dels valors d'entrada varia per a cada cas.

Com ja hem dit repetides vegades, la matriu de commutació dependrà només del nombre de subbandes en que treballem i de la clau d'encriptació aleatòria. Per això la funció '*genmatriu*' té una capçalera en que només li entren aquests dos valors.

```
1   function M = genmatriu(codi,n)
2   %
3   % Genera matriu de commutacio
4   %    codi   codi d'encriptació (tipus PIN)
5   %    n      nombre de subbandes
```

La matriu de commutació resultant ha de tenir les dimensions del nombre de subbandes, ja que és també el nombre de vectors que caldrà permutar. Necessitant així aquest nombre d'entrades i de sortides de la matriu.

```
13 -   M = zeros(n,n);      % matriu de sortida
```

El cos important de la funció és un bucle iteratiu que s'executa per a cada fila i en el qual trobem la posició del '1' pertinent a la fila, eliminant així aquesta posició de cara a l'execució del bucle per a la següent fila, i evitant que repetim més d'un '1' en una columna. La posició la trobem aleatòriament fent una càlculs matemàtics definits a partir del valor del codi d'encriptació i del vector que estiguem tractant en cada iteració, el que ens equival a la fila on es troba aquest.

```
14 -  P = (1:n);          % columnes pendents d'assignar
15
16 -  for i=1:n,          % per cada fila
17 -      div = n+1-i;
18 -      pos = mod(codi,div) + 1;
19 -      j = P(pos);     % determina la columna
20 -      M(i,j)=1;       % assigna valor
21
22 -      j = find(P==j);
23 -      P = [P(1:j-1) P(j+1:div)];    % elimina columna de la llista
24 -      codi = 10000-codi;          % complementa per a aleatoritzar numero
25 -  end
26
```

Podem comprovar que la matriu de commutació depèn única i exclusivament dels dos paràmetres ja esmentats, el nombre de subbandes i la clau d'encriptació. Caldrà però que la clau tingui un valor enter d'entre 0 i 9999, degut a la complementació que efectuem a la funció per aleatoritzar el nombre, però aquest requisit el fixarem en la part d'interfície gràfica, ja que és aleshores quan l'usuari introdueix la clau de forma aleatòria. Malgrat que només treballarà per a 4, 8 o 16 subbandes, la funció és capaç de generar matrius de commutació de qualsevol dimensió.

Així, a la sortida d'aquesta funció n'obtindrem una matriu de commutació especial per al nombre de subbandes i la clau introduïts.

4 4 Reconstrucció del senyal d'àudio

Un cop feta la descomposició del senyal i la permutació de les subbandes, sigui per mesclar les del senyal original, o per recol·locar les del senyal encriptat, caldrà reagrupar aquestes subbandes per obtenir-ne realment un senyal d'àudio.

Anteriorment hem vist quina estructura havia de tenir aquest procés, el de síntesi. Així doncs en cada iteració caldrà delmar totes les subbandes, després filtrar-ne cadascuna per el filtre corresponent, i seguidament sumar-les de dues en dues contiguament. Aquest procés el repetirem les vegades que calgui, depenent del nombre de subbandes que emmagatzema la matriu d'entrada. El nombre de vegades que el repetim és el que anomenem '*etapes*'.

La funció creada s'anomena '*Reconstruct_BandesVaries*', i com queda palès pel seu nom, es dedicarà a reconstruir els senyals que la funció '*BandesVaries*' ens ha descompost en subbandes.

Per tant, la capçalera serà inversa a la de '*BandesVaries*', tenint com a entrades una matriu '*sb*' on tenim les subbandes emmagatzemades com a vectors, la freqüència de mostreig del senyal original '*fm*', i com a sortida un senyal d'àudio que anomenarem '*x*' i que serà un matriu de les dimensions del senyal original.

```
3   function [x] = Reconstruct_BandesVaries(sb,fm)
4                       %% x => Senyal Recontruit/Desencriptat
5                       %% sb => Matriu amb les Subbandes desordenades
6                       %% fm => Freqüència de Mostreig
```

Si aquest funció s'executa en el procés d'encriptació o en el de desencriptació, els continguts de cada una de les variables és diferent. Quan estiguem encriptant, '*sb*' serà el senyal original descompost en subbandes i

amb aquestes desordenades, de tal manera que la '*x*' de sortida serà el senyal d'àudio recompost equivalent al senyal encriptat.

Per al cas de que estiguem en el procés desencriptador, '*sb*' serà el senyal encriptat descompost en subbandes i amb aquestes reordenades, obtenint així a la sortida una '*x*' que sigui, si el procés s'ha fet correctament, la recuperació del senyal d'àudio original.

Un cop iniciada la funció, caldrà inicialitzar els paràmetres necessaris, obtenir aquells que ens puguin ser necessaris per el procés i crear els dos filtres. Recordem que per a obtenir una bona recomposició, seria bo utilitzar dos filtres idèntics als que s'han utilitzat en el procés d'anàlisi. Així, els crearem idèntics, amb les mateixes especificacions.

```
 8    % Creem els 2 filtres, un passa-alt i un passa-baix
 9 -  [Bl,Al]=cheby2(20,80,0.525);% --> Filtre passa-baix
10 -  [Bh,Ah]=cheby2(20,80,0.475,'high');% --> Filtre passa-alt
11
12    % Calculem el tamany que haurà de tenir el senyal de sortida i
13    % i quantes subbandes de frequència tenim en el vector sb
14 -  [m,n] = size(sb);
15            %% m => Tamany de cada subbanda
16            %% n => Nombre de bandes de frequències
17
18    % Inicialitzem el vector de sortida
19 -  x = zeros((n*m),1);
20
21    % Valors inicials de n i m
22 -  nb = n;
23 -  mb = m;
```

Per saber quantes etapes tindrà el procés, cal calcular-ho a partir del logaritme en base 2 del nombre de subbandes.

```
25 -  M = log2(n); % M => nombre de processos depenent del N° de Subbandes
```

Un cop definit el nombre d'etapes, i tots els paràmetres previs al procés, ja entrarem a executar la síntesi.

Entrarem en un bucle que es repetirà tantes vegades com etapes tingui el procés. Així, cada vegada que entrem en aquest bucle, tindrem un nombre de subbandes més reduït fins a arribar a tenir-ne només una, la última, la que és en realitat el senyal recompost.

Cada etapa segueix els mateixos passos, però amb nombre de subbandes diferents. Comencem per inicialitzar els vectors auxiliars que utilitzarem per a fer la interpolació de cada subbanda, necessitant-ne tants com subbandes tinguem en la etapa en que ens trobem. Tot seguit fem aquestes interpolacions amb el corresponent augment d'amplitud de cada subbanda per contrarrestar l'atenuació que ha sofert cadascuna al ser delmada, i això seguit del correponent filtratge de cada subbanda interpolada. Un cop filtrades, les sumem de dues en dues de forma contigua, per reduïr el nobre de subbandes a la meitat, i quan ja en tenim la meitat, ho tornem a emmagatzemar en un vector que prèviament hem inicialitzat. Acabada la etapa on estem, ara entrem a la següent, on treballarem amb la meitat de subbandes, però amb el doble de mostres per cadascuna. Quan arribem a l'última etapa, en que fem totes aquestes operacions per a dues subbandes, un cop interpolades i filtrades, les sumem i d'aquesta suma, que n'obtenim un sol vector, ja en treiem el senyal d'àudio recompost. No cal que ho emmagatzemem enlloc més, ja tenim el senyal que buscàvem.

```
 27 -  for N = 1:M ,
 28
 29 -      if (nb==16)
 30
 31            % Inicialitzem els 16 vector amb que farem l'interpolació
---  -          inicialitzacions
 38            % Per les 16 subbandes fem la interpolació i doblem l'amplitud que
 39            % hem perdut al delmar
---  -          interpolacions
 57            % Filtrem cada subbanda interpolada
---  -          filtratges
 76            % Sumem les subbandes filtrades de dues en dues
---  -          sumes
 87            % Reassignem els valors de sb i de nb, per poder continuar
 88            % la descomposició
---  -          reassignacions
---  -
102 -      elseif (nb==8)
---  -
---  -          repetim inicialitzacions, interpolacions, filtratges,
---  -          sumes i reassignacions per a 8 subbandes
---  -
148 -      elseif (nb==4)
---  -
---  -          repetim inicialitzacions, interpolacions, filtratges,
---  -          sumes i reassignacions per a 4 subbandes
---  -
180 -      elseif (nb==2)
---  -
---  -          repetim inicialitzacions, interpolacions i
---  -          filtratges per a 2 subbandes
---  -
195            % sumem les dues subbandes filtrades
196 -          x_temp = xl + xh; % --> Senyal Reconstruit
197
198 -      end
199
200 -  end
201
202 -  x = x_temp; % --> Senyal Reconstruit
203
204
```

El bucle es repetirà tantes vegades com etapes tinguem. Per a 16 subbandes, el nombre d'etapes serà de 4, per tant, s'executaran primer les intruccions per *nb=16*, després per *nb=8*, després per *nb=4* i finalment per *nb=2*, abtenint-ne al final el senyal recompost. Si per el contrari, el nombre de subbandes inicial és de 2, tindrem només una etapa en la qual executarem

només les intruccions per *nb=2* i ja haurem acabat. I així per als quatre possibles casos.

Les inicialitzacions que hem de fer en cada cas són d'aquells vectors auxiliars que ens ajudaran a fer la interpolació. Per el cas de quatre subbandes, serien aquestes inicialitzacions:

```
150          % Inicialitzem els 4 vector amb que farem l'interpolació
151 -        auxc1 = zeros(mb*2,1);
152 -        auxc2 = auxc1; auxc3 = auxc1; auxc4 = auxc1;
```

Amb aquests vectors auxiliars, ara és quan es fa la interpolació amb el degut augment d'amplitud de cada subbanda. Recordem que al delmar el que hem fet ha estat reduïr el nombre de mostres a la meitat escollint-ne una de cada dues, i per tant, l'amplitud de cada subbanda s'ha vist reduida a la meitat. Per això ara caldrà augmentar-ne l'amplitud. Aquest augment es materialitza multiplicant el valor de cada mostra que conforma el vector de la subbanda per dos. Seguirem mostrant el procés per a quatre subbandes, per veure'n la continuitat i el seu sentit:

```
154          % Per les 4 subbandes fem la interpolació i doblem l'amplitud que
155          % hem perdut al delmar
156 -        auxc1(1:2:length(auxc1)) = ((sb(1:mb,1))*2);    %%Baixes freqs
157 -        auxc2(1:2:length(auxc2)) = ((sb(1:mb,2))*2);
158 -        auxc3(1:2:length(auxc3)) = ((sb(1:mb,3))*2);
159 -        auxc4(1:2:length(auxc4)) = ((sb(1:mb,4))*2);    %%Altes freqs
```

La interpolació és l'augment de mostres d'un senyal per aproximar-nos al valor real del senyal. Així, una interpolació per dos, vol dir que entre mostra i mostra d'un senyal, hi afegim mostres. Poden tenir el valor amitjanat de les dues mostres entre les quals es col·loca la nova, o pot tenir el valor d'una d'elles, o simplement ser una mostra buida, amb valor *zero*. En el nostre cas, hem escollit interpolar de tal manera que cada nova mostra serà *zero*, tenint en

compte que a cada mostra del vector li hem de doblar d'amplitud, amb la qual cosa ja quedarà la amplitud global dessitjada del senyal que aquest vector representa. Per poder fer-ho així, hem fet els dos passo acabats d'exposar. Enlloc d'agafar el vector de mostres i introduïr-hi entre les mostres una mostra buida, ho hem fet al revés. Hem creat els vectors auxiliars que acabem de veure, amb un temany el doble que cada subbanda, i sent una cadena de *zeros*. I tot seguit li hem substituït una de cada dues mostres buides per una mostra del vector que ens representa la subbanda pertinent, amb la amplitud degudament doblada.

Seguint amb la síntesi del senyal, ara és el moment de filtrar. Cada subbanda ja interpolada, la filtrem per el corresponent filtre (el pass-alt o el passa-baix). La utilització d'un filtre o un altre queda definit per la situació de la subbanda. Recordem la estructura d'arbre de la síntesi (*Figura3.11*) i l'anàlisi (*Figura3.9*).

```
161          % Filtrem cada banda interpolada
162 -        xll = filter (Bl,Al,auxc1); %% Baixes freqs
163 -        xlh = filter (Bh,Ah,auxc2);
164 -        xhl = filter (Bl,Al,auxc3);
165 -        xhh = filter (Bh,Ah,auxc4); %% Altes freqs
```

Un cop situada cada subbanda on li pertoca, gràcies a la interpolació i al filtratge, ara en toca sumar les dues subbandes contígües. En aquest cas que estem exposant, hauríem de sumar les dues subbandes d'altes freqüències per un costat i les dues de baixes per un altre.

```
167          % sumem les bandes filtrades de dues en dues
168 -        xl = xll + xlh; %% Baixes freqs
169 -        xh = xhl + xhh; %% Altes freqs
```

Ara ja tenim les quatre subbandes d'aquesta etapa en dos vector que ens representen les dues subbandes principals del senyal d'àudio. Com que encara no acabem, segons el bucle de les etapes, encara ens en queda una última, cal que emmagatzemem aquestes dues subbandes que tenim per poder entrar de

nou al bucle i repetir els processos que acabem de descriure, però aquest cop per dues subbandes enlloc de quatre. Així, reinicialitzem el vector '*sb*' amb les dimensions de les dues subbandes que tenim ara, i tornem a obtenir el valor de '*nb*', el que ens permetrà que en la nova i última etapa entrem en el cas de *nb=2* per poder seguir amb el procés de síntesi fins el final.

```
171          % Reassignem els valors de sb i de nb, per poder continuar
172          % la descomposició
173 -        sb = zeros(length(xl),2);% --> Reinicialitzem 'sb'
174
175 -        sb(1:length(xl),1) = xl;    %% Baixes freqs
176 -        sb(1:length(xl),2) = xh;    %% Altes freqs
177
178 -        [mb,nb] = size(sb);% --> nou valor de 'nb'
```

Un cop fets aquests reassignaments de valors, tornarem a entrar al bucle per acabar amb la última etapa.

Els processos acabats de mostrar són per la etapa en que tenim 4 subbandes. Si en tenim més, és a dir 8 o 16, els processos són idèntics, però amb el nombre de subbandes i per tant d'operacions, doblades. Farem el doble d'inicialitzacions, el doble d'interpolacions, el doble de filtratges, el doble de sumes i la reinicialització de '*sb*' amb el doble de subbandes. Però conceptualment parlant, els processos són idèntics per a les quatre possibles etapes. Només canvia el final de la etapa de 2 subbandes, que com ja s'ha comentat al principi d'aquest apartat, un cop sumades les dues subbandes que s'han interpolat i filtrat, el vector resultant ja és el final, per tant, no caldrà fer reassignacions del vector '*sb*' ni del valor de '*nb*'. Simplement sortim de la etapa de dues subbandes i del bucle d'etapes, i treiem el valor final de l'últim vector obtingut, el que serà el senyal d'àudio reconstruït, i ho assignem al vector de sortida de la funció.

```
195              % sumem les dues subbandes filtrades
196 -            x_temp = xl + xh; % --> Senyal Reconstruit
197
198 -        end
199
200 -    end
201
202 -    x = x_temp; % --> Senyal Reconstruit
```

5 Interfície gràfica

Un dels objectius d'aquest projecte, a part del bon funcionament del sistema encriptador/desencriptador creat, també era aconseguir crear una bona interfície gràfica. Entenent per 'bona' que la interfícies sigui senzilla, útil i fàcil d'usar, que d'una manera molt intuïtiva qualsevol persona pugui utilitzar-la, i alhora ens mostri els resultats de tal manera que es pugui comprendre què és tot el que ha succeït al llarg del procés d'encriptació i de desencriptació.

5 1 Funcionament general

La interfície gràfica creada s'ha fet mitjançant l'eina Guide del mateix programa Matlab, el que ens ha permès una integració perfecta entre el codi del sistema encriptador/desencriptador i el de la mateixa interfície.

La interfície, a trets generals, ens permet escollir un arxiu d'àudio del tipus 'wav' emmagatzemat en l'ordinador, i veure'n les seves característiques. Ens mostra la gràfica del senyal al llarg del temps i el seu espectre, així com un botó amb la possibilitat d'escoltar-lo. Després ens permet escollir el nombre de subbandes en que es vol treballar, sempre entre 2, 4, 8 o 16, i introduir-hi una clau d'encriptació, d'entre 0 a 9999. Un cop escollits els dos valors correctament, ens permet encriptar el senyal d'àudio. I si l'encriptem, seguidament en torna a mostrar totes les característiques del senyal encriptat. Senyal al llarg del temps, el seu espectre, i la possibilitat d'escoltar-lo i constatar així que el procés d'encriptació funciona correctament i el senyal ha perdut tota la seva intel·ligibilitat, sense perdre però informació. En aquest punt tornem a tenir la possibilitat d'escollir un nombre de subbandes i una clau d'encriptació, aquest cop per el procés de desencriptació, i un cop escollits, podem provar de desencriptar el senyal. Si la clau introduïda i el nombre de subbandes han coincidit amb els primers, la informació que ara ens mostrarà la interfície hauria de ser molt semblant a la mostrada en el primer moment, en

llegir l'arxiu original. Tornem a tenir el senyal representat al llarg del temps, també la seva distribució espectral, i ara ens dona la possibilitat, mitjançant tres botons diferents, d'escoltar el senyal original, l'encriptat i el desencriptat, podent així comparar auditivament els tres senyals i veure si la pèrdua de d'informació deguda al filtratge que veiem en la gràfica realment és important auditivament parlant. Si per el contrari els dos valors introduït per desencriptar no coincideixen amb els de la encriptació, tant en les representacions gràfiques, com en els botons d'àudio, podrem constatar clarament que el senyal recuperat no és pas l'original, veient demostrat també que aquest és un bon sistema d'encriptació d'àudio. La interfície també consta d'un botó que permet netejar tot el que s'ha fet i tornar a començar de nou, encara que igualment si algú canvia els valor de subbandes o clau a mig procés, aquest torna en el punt de lectura d'aquests valors, esborrant les gràfiques i els senyals d'àudio aconseguits fins aleshores.

Si s'ha fet tot correctament, l'aparença de la interfície gràfica al final del procés pot ser la següent:

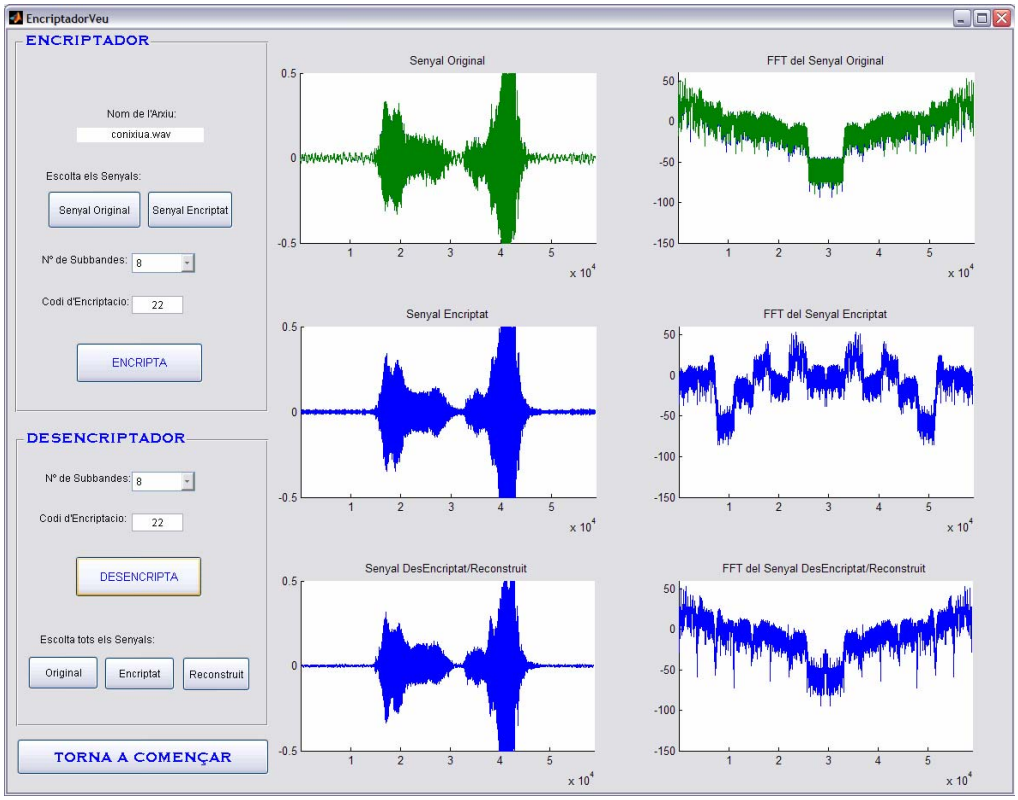

Figura5.1

Fixant-nos en els espectres de la *Figura5.1*, podem veure com el segon, l'encriptat, és molt diferent al primer, l'original, i en canvi el tercer, el desencriptat, és gairebé igual, exceptuant petites pèrdues degudes al filtratge, però que no en danyen la intel·ligibilitat, ja que la distribució espectral és molt similar.

Si no es tenen els dos valors correctes, nombre de subbandes i clau d'encriptació aleatòria, és impossible recuperar el senyal d'àudio original, i per tant, desxifrar-lo. En el cas anterior, en que s'havia encriptat amb 8 subbandes i una clau d'encriptació '22', si intentem recuperar el senyal però ens equivoquem en un dels dos paràmetres, per exemple, la clau la posem de '23' enlloc de '22', el senyal recuperat és inintel·ligible.

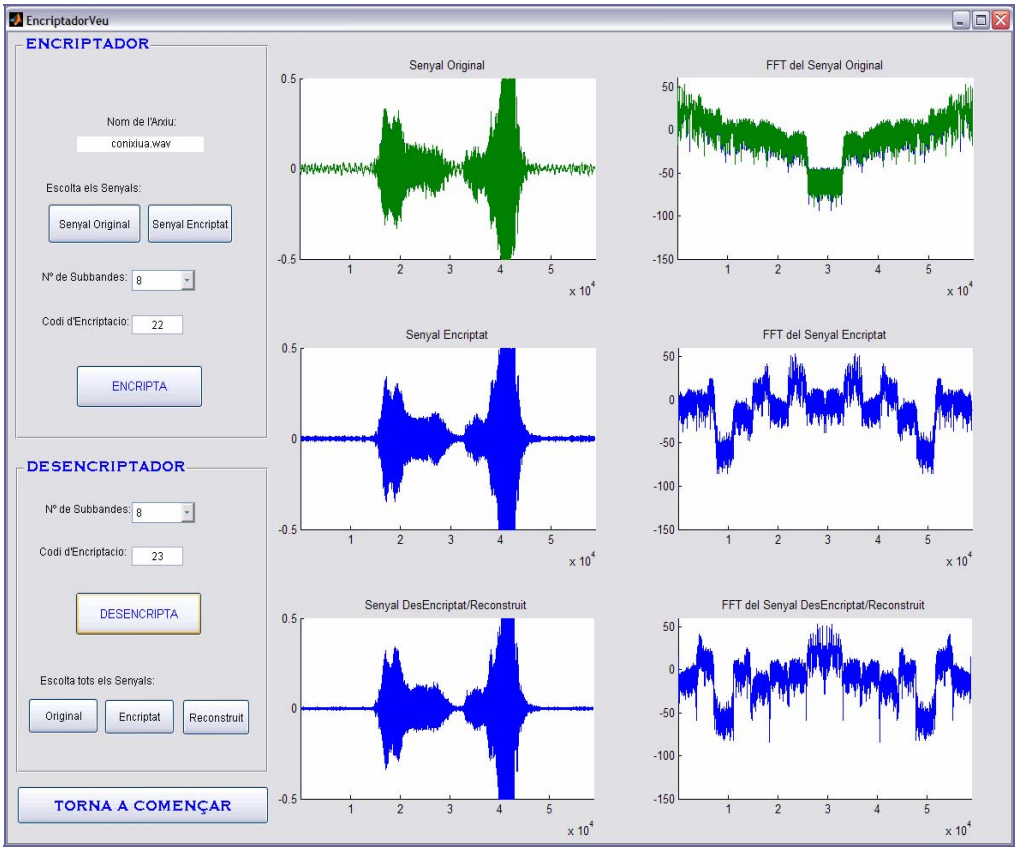

Figura5.2

Altre cop mirant l'espectre, queda palès que el senyal que s'ha intentat recuperar no s'assembla en res a l'original. L'error en la clau ha fet que la matriu de commutació al desencriptar no fos la transposada exacta de la que

hem usat en encriptar, i això ha fet que les subbandes es tornessin a permutar, però altra vegada situant-se en posicions errònies de l'espectre.

5 2 Implementació en Matlab

La implementació de la interfície gràfica s'ha dut a terme amb l'eina *Guide* del propi programa *Matlab*.

Guide és una aplicació que combina dos mètodes de treball. L'un és gràficament, per on definirem els botons, la seva posició i les seves característiques 'estètiques', així com les finestres per on veurem les gràfiques, etc. L'altra és de codi. En algun lloc hem de definir què és el que farà cada element de la interfície, si cridarà a una funció, si serà una entrada o sortida de dades, etc. També amb aquest mètode serà com definirem quan un element de la interfície apareix o es manté ocult, o quan passa a ser operatiu.

Entrarem a comentar les set funcions de que consta la part de codi de la interfície, ja que la part gràfica la podem observar directament. Ho farem seguint l'ordre d'execució del programa, per apreciar-ne bé el seu funcionament.

5 2 1 Programa principal

És el programa principal on, al definir gràficament les característiques físiques de tots els elements de la interfície, el programa mateix ens crea les línies de codi pertinents on queden especificades totes aquestes característiques inicials del elements.

La nostra funció pertinent al programa principal s'anomena '*EncriptadorVeu*', però com acabem de comentar, la crea el mateix *Guide*, per tant, no entrarem a comentar les línies de codi, ja que aquestes línies contenen les especificacions dels element que directament podem observar en la mateixa interfície.

5 2 2 Selecció de l'arxiu a encriptar

Aquesta és la funció que anomenem '*sel_arxiu*', i serà la primera que s'executi. Això és degut a que en un bon començament hem definit que la interfície gràfica només ens mostri un dels botons, l'anomenat '*ESCULL ARXIU*' ocultant-nos tota la resta d'elements que la conformen, i aquest botó al clicar-lo té la ordre de cridar a aquesta funció '*sel_arxiu*'. Per tant, inevitablement, aquesta serà la primera funció que invoquem.

Un cop invocada la funció, el primer que fa és ocultar el botó d''*'ESCULL ARXIU*', per evitar que l'usuari provi d'obrir un nou arxiu a mig procés, sense haver-lo aturat abans.

```
6     % Amaguem el botó per escollir arxiu
7 -   H1 = findobj(gcf,'Tag','h1');
8 -   set(H1,'Visible','off');
```

A continuació s'executa la funció del mateix Matlab '*uigetfile*', la qual ens obre un quadre d'exploració de l'ordinador en busca d'arxius d'àudio del tipus 'wav'.

```
13    % Funció que ens obre un quadre de diàleg per navegar per l'ordinador i
14    % escollir l'arxiu *.wav dessitjat
15 -  [nom , carpeta] = uigetfile('*.wav', 'Fitxers de veu');
```

Un cop s'escull l'arxiu 'wav' desitjat, l'hem de llegir, mitjançant la funció '*wavread*' també del propi Matlab, obtenint-ne el o els vectors que conformen el senyal d'àudio (un o dos vectors depenent de si és *mono* o *estèreo* respectivament), i la seva freqüència de mostreig.

```
17      % Funció de lectura del fitxer *.wav escollit
18 -    [x,fm] = wavread( nom );
```

Al tenir ja l'arxiu d'àudio seleccionat, ja podrem encriptar, per tant, activem tots aquells elements que necessitem.

```
22      % Ens apareixeran per pantalla els seguents 'botons'
23      % Nom de l'arxiu escollit
24 -    HSE = findobj(gcf,'Tag','hse');
25 -    set(HSE,'Visible','on');
26      % Apareix el nom de l'arxiu per pantalla
27 -    set(HSE,'String',nom);
28      % Cartell de "Nom Arxiu"
29 -    HS1 = findobj(gcf,'Tag','hs1');
30 -    set(HS1,'Visible','on');
31      % Selecció del Codi d'encriptació
32 -    H2 = findobj(gcf,'Tag','h2');
33 -    set(H2,'Visible','on');
34      % Cartell "Codi:"
35 -    H21 = findobj(gcf,'Tag','h21');
36 -    set(H21,'Visible','on');
37      % Botó per fer sonar l'arxiu original
38 -    H4 = findobj(gcf,'Tag','h4');
39 -    set(H4,'Visible','on');
40      % Cartell "Escolta els senyals"
41 -    H44 = findobj(gcf,'Tag','h44');
42 -    set(H44,'Visible','on');
43      % Cartell 'Torna a començar'
44 -    HNET = findobj(gcf,'Tag','netejar');
45 -    set(HNET,'Visible','on');
```

Abans de fer visible els nombre de subbandes possibles que l'usuari podrà escollir, cal determinar quines possibilitats hi ha. Depenent de la freqüència de mostreig de senyal d'entrada '*fm*', podrem fer les operacions per un nombre de subbandes o no. Seguint la mateixa taula de freqüències explicada anteriorment (*Figura4.1* - apartat **4 2**), alhora d'activar aquest camp,

haurem de veure primer si cal limitar-ho o no. Un cop decidit, també mostrarem el cartell '*Nº Subbandes*'.

```
47    % Selecció del nombre de subbandes depenent de la fm i aparició per pantalla
48 -      if (fm<10000)% Si és menor de 10kHz ho limitem a 2 o 4 subbandes
49 -          H3 = findobj(gcf,'Tag','h3');
50 -          set(H3,'String','Escull|2|4');
51 -          set(H3,'Visible','on');
52 -          HH3 = findobj(gcf,'Tag','hh3');
53 -          set(HH3,'String','Escull|2|4');
54 -      elseif ((fm>10000)&(fm<30000))% Si està entre 10KHz i 30KHz podem amb 8
55 -          H3 = findobj(gcf,'Tag','h3');
56 -          set(H3,'String','Escull|2|4|8');
57 -          set(H3,'Visible','on');
58 -          HH3 = findobj(gcf,'Tag','hh3');
59 -          set(HH3,'String','Escull|2|4|8');;
60 -      elseif (fm>30000)% Major de 30KHz ens permet fer-ho entre 2, 4, 8 o 16
61 -          H3 = findobj(gcf,'Tag','h3');
62 -          set(H3,'String','Escull|2|4|8|16');
63 -          set(H3,'Visible','on');
64 -          HH3 = findobj(gcf,'Tag','hh3');
65 -          set(HH3,'String','Escull|2|4|8|16');
66 -      end
67
68    % Aparició per pantalla del Cartell "Nº Subbandes:"
69 -  H31 = findobj(gcf,'Tag','h31');
70 -  set(H31,'Visible','on');
```

Un cop ja tenim tots el elements que ens interessen actius i presents en la interfície, ara representarem gràficament el senyal d'àudio que hem escollit. En farem la representació temporal i la representació espectral, aquest última a partir del càlcul de la FFT del senyal.

```
73    % Expressem gràficament el resultat obtingut
74    % Representació del Senyal Original per la finestra 'axes1'
75 -  AX1 = findobj(gcf,'Tag','axes1');
76 -  set(AX1,'Visible','on');
77 -  set(gcf,'CurrentAxes',AX1);
78 -  ax1 = get(gcf,'CurrentAxes');
79 -  plot(ax1,x); axis([1 length(x) -0.5 0.5])
80 -  title('Senyal Original');
81
82    % Representació de la FFT del Senyal Original per la finestra 'axes2'
83 -  X = 20*log10(abs(fft(x)));% --> Càlcul de la FFT
84 -  AX2 = findobj(gcf,'Tag','axes2');
85 -  set(AX2,'Visible','on');
86 -  set(gcf,'CurrentAxes',AX2)
87 -  ax2 = get(gcf,'CurrentAxes');
88 -  plot(ax2,X); axis([1 length(X) -150 60])
89 -  title('FFT del Senyal Original');
```

Al finalitzar tota la execució d'aquest funció la interfície gràfica ja tindrà una altra aparença, ja que s'han fet visibles alguns dels elements que la componen, així com les gràfiques del senyal original, les corresponents al senyal al llarg del temps i la del seu espectre freqüencial.

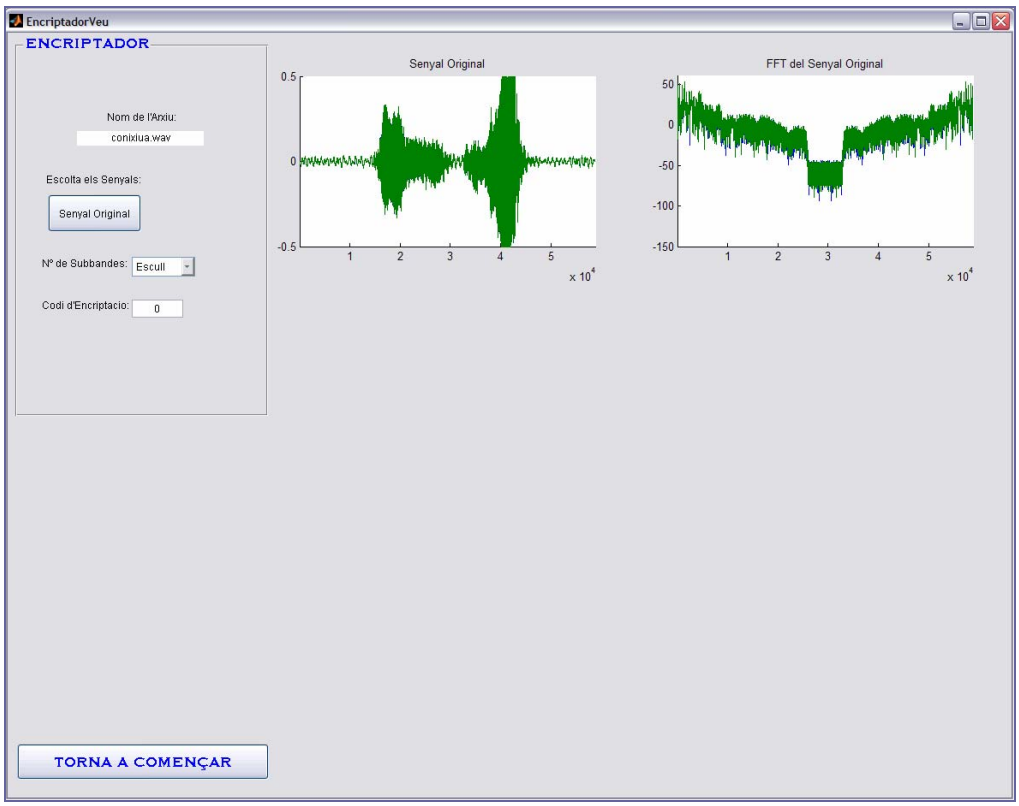

Figura5.3

Un dels botons que apareix és el que s'anomena '*Senyal Original*', que tal com n'indica el títol, és un botó que ens permetrà escoltar el senyal d'àudio que acabem d'escollir i de representar. Aquest botó està associat a la funció de Matlab '*sound()*', la que ens permet escoltar senyals d'àudio.

A part d'aquest botó per escoltar el senyal, la única opció que l'usuari pot fer és escollir el nombre de subbandes en que vol encriptar el senyal i la clau d'encriptació que vol utilitzar. És en aquest punt en que saltem a la següent funció, la funció 'dades', la que ens permetrà obtenir les dades que l'usuari introdueixi.

5 2 3 Obtenció de les dades per encriptar

Automàticament, quan l'usuari introdueixi una de les dues dades especificades, sigui el nombre de subbandes o el codi d'encriptació, entrarem a la funció d'obtenció de dades, l'anomenada '*dades*'.

La funció, com el seu nom indica, s'encarregarà d'obtenir les dades que l'usuari hagi introduït.

Primer, obtenim el codi d'encriptació, que ens entra com una cadena de caràcters, però que amb la funció '*str2num*' passarem a valor enter per poder més endavant treballa-hi com a tal.

```
3     % Obtenció de número de Codificació
4 -   H2 = findobj(gcf,'Tag','h2');
5 -   g = str2num(get(H2,'String'));
```

I seguidament obtenim el nombre de subbandes.

```
 7     % Obtenció del número de Subbandes dessitjat per l'usuari
 8 -   H3 = findobj(gcf,'Tag','h3');
 9 -   dada = get(H3,'Value');
10
11    % Assignació del nombre de Subbandes en relació al valor escollit
12 -   if(dada==2)
13 -       n = 2;
14 -   elseif(dada==3)
15 -       n = 4;
16 -   elseif(dada==4)
17 -       n = 8;
18 -   elseif(dada==5)
19 -       n = 16;
20 -   end
```

Com la interfície gràfica presenta a l'usuari les opcions que té alhora de triar el nombre de subbandes en forma de llista, quan aquest escull el que n'obtenim és la posició de la llista que ha escollit, no pas el valor que en la llista surt representat i que l'usuari vol. Per tant, cal que dins el programa definim quin valor de subbandes '*n*' vol l'usuari en relació a la posició de la llista que ens ha escollit.

Al final d'aquesta funció de captura de dades, tenim una altra condició, la qual no permetrà que l'usuari pugui iniciar la encriptació si encara no ha seleccionat els dos paràmetres que se li permeten correctament.

És a dir, el botó que permet iniciar la encriptació i que s'anomena '*ENCRIPTA*' no el mostrarem per pantalla fins que tinguem un nombre de subbandes diferent a *0* i una clau d'encriptació amb valor entre 1 i 9998, ambdós inclosos.

```
 98    % Entrarem al seguent 'if' només quan haguem escollit un Codi entre 0 i
 99    % 9999 i N°de Subbandes
100    % Fins que no haguem entrat el N° de Subbandes i el Codi d'Encriptació
101    % no tindrem la possibilitat d'Encriptar el Senyal
102 -  if((g>0)&(g<9999)&(dada>1))
103
104        % Apareixeran per pantalla el Botó "ENCRIPTA"
105 -      H6 = findobj(gcf,'Tag','h6');
106 -      set(H6,'Visible','on');
107
108        % El Contador ens marca el punt de l'Execució on ens trobem
109 -      contador=2;
110
111 -  end
```

El comptador que veiem és per saber en quin punt de l'execució ens trobem, per si l'usuari vol tornar a començar. En totes les funcions on fem aparèixer elements de la interfície que estaven ocults tenim comptadors que permetran que la funció '*neteja*' associada al botó '*TORNA A COMENÇAR*' sàpiga quins elements de la interfície estan a la vista i quins encara no, i per tant torni a amagar només aquests elements vistos, sense haver-se de preocupar per els que s'haurien activat més endavant en el cas d'haver seguit amb la execució.

Ara la interfície ens ha quedat molt similar a la que ja teníem, amb la diferència que ara tenim nombre de subbandes i clau d'encriptació escollits, i sobretot amb tenim a la vista el botó '*ENCRIPTA*', el que ens permetrà tirar endavant amb el procés, és a dir, saltar a la funció '*encripta*', la següent.

Figura5.4

5 2 4 Procés d'encriptació

El botó '*ENCRIPTA*' fa la crida a la funció del mateix nom '*encripta*'.

Aquesta funció principalment fa la crida al programa principal del nostre sistema encriptador, és a dir, a la funció '*Encriptador_ok*'. Aquí és on entra clarament la interacció entre la interfície i els nostre encriptador de veu.

```
4 -   y = Encriptador_ok(x,g,n,fm);
5             %% y => Senyal Encriptat
6             %% x => Senyal Original
7             %% g => Codi d'Encriptació
8             %% n => Nº de Subbandes escollit
9             %% fm => Freqüència de Mostreig del Senyal Original
```

Com veiem els paràmetres d'entrada són els que hem anat obtenint prèviament, primer amb la funció '*sel_arxiu*' d'on n'hem tret el senyal d'àudio original '*x*' i la seva freqüència de mostreig '*fm*', i després amb la funció '*dades*' on hem obtingut el nombre de subbandes '*n*' i la clau d'encriptació '*g*'.

Un cop s'hagi executat tot el programa d'encriptació, tindrem el senyal encriptat '*y*', i caldrà mostrar-ne les característiques a l'usuari, perquè aquest pugui apreciar el funcionament de l'encriptador. Així doncs, activem el botó '*Senyal Encriptat*' que permetrà escoltar-lo, i fem les dues representacions gràfiques, la del senyal al llarg del temps i la de la seva distribució espectral.

```
11    % Fem visible el botó per fer sonar el Senyal Encriptat
12 -  H5 = findobj(gcf,'Tag','h5');
13 -  set(H5,'Visible','on');
14
15    % Representació del Senyal Encriptat en la finestra 'axes3'
16 -  AX3 = findobj(gcf,'Tag','axes3');
17 -  set(AX3,'Visible','on');
18 -  set(gcf,'CurrentAxes',AX3);
19 -  ax3 = get(gcf,'CurrentAxes');
20 -  plot(ax3,y); axis([1 length(y) -0.5 0.5])
21 -  title('Senyal Encriptat');
22
23    % Representació de la FFT del Senyal Encriptat en la finestra 'axes4'
24 -  Y = 20*log10(abs(fft(y)));% --> Càlcul de la FFT
25 -  AX4 = findobj(gcf,'Tag','axes4');
26 -  set(AX4,'Visible','on');
27 -  set(gcf,'CurrentAxes',AX4);
28 -  ax4 = get(gcf,'CurrentAxes');
29 -  plot(ax4,Y); axis([1 length(Y) -150 60])
30 -  title('FFT del Senyal Encriptat');
```

També caldrà que mostrem els elements de la interfície que permetran intentar desencriptar el senyal que acabem de xifrar.

```
32      % Fem visibles els botons del Desencriptador
33      % Panell del 'DESEMCRIPTADOR'
34 -    PAN2 = findobj(gcf,'Tag','uipanel2');
35 -    set(PAN2,'Visible','on');
36      % Elecció del nombre de Subbandes
37 -    HH3 = findobj(gcf,'Tag','hh3');
38 -    set(HH3,'Visible','on');
39      % Cartell 'Nombre de Subbandes'
40 -    HH31 = findobj(gcf,'Tag','hh31');
41 -    set(HH31,'Visible','on');
42      % Codi de Desencriptació
43 -    HH2 = findobj(gcf,'Tag','hh2');
44 -    set(HH2,'Visible','on');
45      % Cartell 'Codi'
46 -    HH21 = findobj(gcf,'Tag','hh21');
47 -    set(HH21,'Visible','on');
```

Ara estem en el punt en que hem finalitzat tot el procés d'encriptació del senyal d'entrada i la aparença de la interfície gràfica comença a complicar-se. Ara ja tenim dos senyals representats, amb els respectius botons per escoltar-los, i amb les especificacions d'encriptació que l'usuari ha introduït. I ja tenim també el quadre de característiques del procés de desencriptació a punt per intentar-ho.

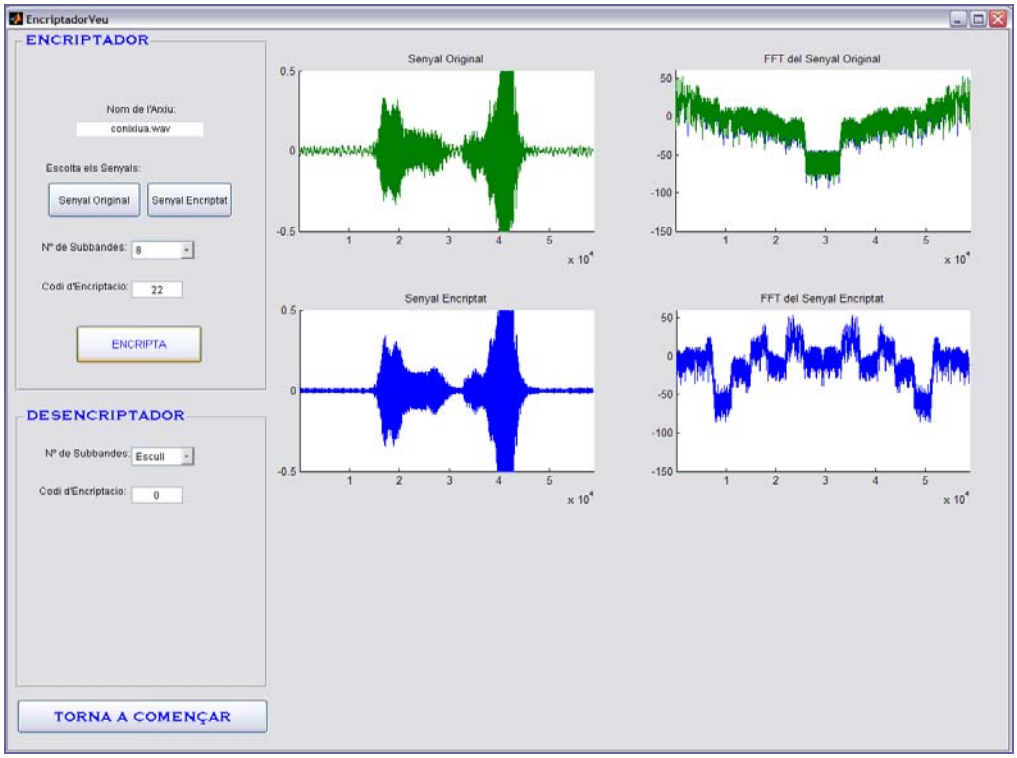

Figura5.5

Ara tenim la possibilitat de començar el procés de desencriptació, i per poder tirar endavant caldrà que definim les característiques de la desencriptació, és a dir, el nombre de subbandes i la clau de desencriptació. Aquest cop, quan definim aquests paràmetres, saltarem directes a la funció '*dades_des*' que ens llegirà aquests paràmetres.

5 2 5 Obtenció de les dades de desencriptació

Quan s'introdueixi un dels os paràmetres de configuració del procés de desencriptació, entrarem a la funció d'obtenció d'aquestes dades, la funció '*dades_des*'.

Aquesta funció serà idèntica a la d'obtenció de les dades d'encriptació ('*dades*'), amb la única diferència que aquests paràmetres de configuració els llegirem en un altre element de la interfície, és a dir, la lectura de les dades la

farem en els valors d'entrada del panell '*DESENCRIPTADOR*', i no del '*ENCRIPTADOR*' com en la funció '*dades*'.

Així, les línies de codi no les mostrarem, ja que són una repetició de les de la funció '*dades*' (*apartat **5 2 3***), però sí que farem una ullada a com ens queda la interfície després d'aquesta funció.

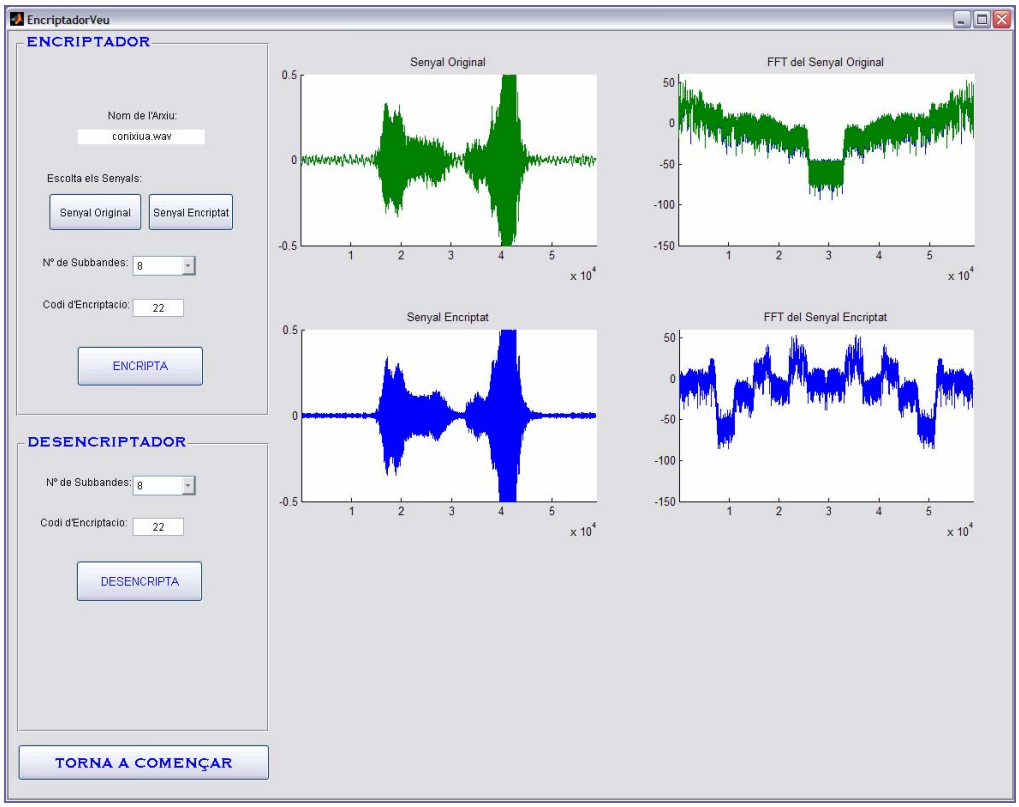

Figura5.6

Com anteriorment succeïa amb la funció '*dades*', ara ens ha aparegut el botó que permetrà tirar endavant el procés, en aquest cas, el botó '*DESENCRIPTA*' que ens permetrà intentar desencriptar el senyal que tenim encriptat.

5 2 6 Procés de desencriptació

El procés s'iniciarà quan l'usuari cliqui en el botó '*DESENCRIPTA*', el qual està associat a la funció homònima, '*desencripta*'.

La funció '*desencripta*' començarà per fer la crida i la execució del programa principal del nostre sistema desencriptador, la funció '*DesEncriptador_ok*'. Només en un cas no s'executarà aquesta funció, concretament si pretenem fer la desencriptació amb dues subbandes i comprovem de bon començament que la clau d'encriptació '*g*' (obtinguda en la funció '*dades*') i la clau de desencriptació '*gg*' (obtinguda en la funció '*dades_des*') no coincideixen. Recordem que és un requisit indispensable per poder recuperar correctament un senyal encriptat que tant el nombre de subbandes en que es fan els dos processos com les dues claus, siguin idèntiques. Per el cas de dues subbandes però codis diferents ens estalviarem entrar en el programa principal de desencriptació, facilitant així el funcionament del programa. I és que si entréssim en el programa principal amb aquestes característiques, perdríem temps d'execució per al final obtenir-ne el mateix resultat, és a dir, un senyal a la sortida que continua estant encriptat. Nosaltres el que farem és que per aquest cas aïllat directament assignarem al vector de sortida, per on hauríem de treure'n el senyal original recuperat, el vector del senyal encriptat.

```
4    % Si encriptem amb 2 Subbandes i els codis d'Encriptació i Desencriptació
5    % NO coincideixen, el senyal reconstruit serà el mateix que l'encriptat.
6    % Per a qualsevol altre cas, executarem la funció 'DesEncriptador_ok'.
7 -  if((n==2)&(g~=gg))
8 -      xx = y;
9 -  else
10 -     xx = DesEncriptador_ok(y,gg,nn,fm);
11               %% xx => Senyal Desencriptat
12               %% y => Senyal Encriptat
13               %% gg => Codi de Desencriptació
14               %% nn => Nº de Subbandes alhora de Desencriptar
15               %% fm => Frequència de Mostrig del Senyal Original
16 -  end
```

Recordem que els paràmetres d'entrada de la funció '*DesEncriptador_ok*' són el senyal original encriptat '*y*', la freqüència de mostreig del senyal original '*fm*', i els dos paràmetres introduïts per l'usuari en el panell '*DESENCRIPTADOR*' i obtinguts amb la funció '*dades_des*', el nombre de subbandes en que desencriptarem '*nn*' i la clau de desencriptació '*gg*'. La sortida del desencriptador '*xx*' serà la reconstrucció del senyal original si les claus de encriptació i desencriptació, i els nombres de subbandes en un procés i en l'altre, coincideixen respectivament. En el cas de que uns o altres paràmetres no siguin idèntics, el senyal '*xx*' continuarà sent un senyal encriptat.

Per poder apreciar el funcionament del nostre sistema encriptador/ desencriptador hem creat tres botons que en aquest punt mostrarem per pantalla, els quals ens permeten escoltar els tres senyals del procés, és a dir, el senyal original, el senyal encriptat i el senyal desencriptat o reconstruït. Aquests botons duen precisament aquests tres noms, '*Senyal Original*', '*Senyal Encriptat*' i '*Senyal Reconstruït*'.

```
18      % Fem visibles els seguents botons
19      % Cartell 'Escolta els Senyals'
20 -    HH44 = findobj(gcf,'Tag','hh44');
21 -    set(HH44,'Visible','on');
22      % Botó per escoltar el Senyal Original
23 -    HH4 = findobj(gcf,'Tag','hh4');
24 -    set(HH4,'Visible','on');
25      % Botó per escoltar el Senyal Encriptat
26 -    HH5 = findobj(gcf,'Tag','hh5');
27 -    set(HH5,'Visible','on');
28      % Botó per escoltar el Senyal Reconstruit
29 -    HH7 = findobj(gcf,'Tag','hh7');
30 -    set(HH7,'Visible','on');
```

Finalment caldrà que mostrem els senyal desencriptat gràficament, mitjançant la seva representació temporal i la seva representació espectral.

```
32      % Representació del Senyal Desencriptat en la finestra 'axes5'
33 -    AX5 = findobj(gcf,'Tag','axes5');
34 -    set(AX5,'Visible','on');
35 -    set(gcf,'CurrentAxes',AX5);
36 -    ax5 = get(gcf,'CurrentAxes');
37 -    plot(ax5,xx); axis([1 length(y) -0.5 0.5])
38 -    title('Senyal DesEncriptat/Reconstruit');
39
40      % Representació de la FFT del Senyal Desencriptat en la finestra 'axes6'
41 -    XX = 20*log10(abs(fft(xx)));% --> Càlcul de la FFT
42 -    AX6 = findobj(gcf,'Tag','axes6');
43 -    set(AX6,'Visible','on');
44 -    set(gcf,'CurrentAxes',AX6);
45 -    ax6 = get(gcf,'CurrentAxes');
46 -    plot(ax6,XX); axis([1 length(Y) -150 60])
47 -    title('FFT del Senyal DesEncriptat/Reconstruit');
```

Ara ja podem considerar que estem a la interfície gràfica definitiva, la que ens mostra els tres passos per els quals el senyal que hem escollit al principi del procés ha anat passant. Tenim les representacions temporals i espectrals del senyal original, del senyal encriptat i del senyal desencriptat, així com els tres botons esmentats anteriorment que ens permeten escoltar els tres senyals. També veiem el nom de l'arxiu del senyal original, i els paràmetres que s'han configurat en la encriptació i en la desencriptació, nombres de subbandes i claus.

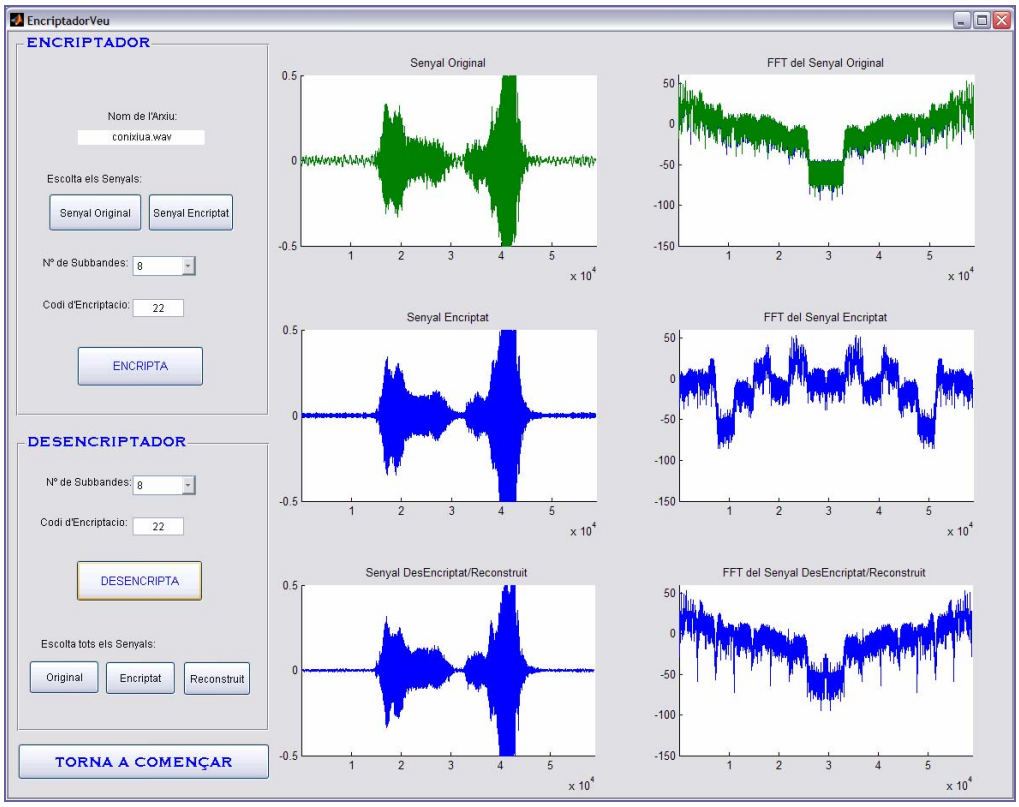

Figura5.7

Omnipresentment tenim en la interfície gràfica un botó anomenat '*TORNA A COMENÇAR*' que com el seu nom indica, ens permet tornar a començar de nou tot el procés d'encriptació/desencriptació.

5 2 7 Reinici dels càlculs

Clicant sobre el botó '*TORNA A COMENÇAR*' cridem a la última funció del programa, la funció '*neteja*'.

És una funció que com el seu nom ja ens diu es dedica a netejar la interfície gràfica. Amb això volem dir que és una funció que ens permetrà tornar a l'inici del tot sense necessitat de tancar l'aplicació i tornar-la a obrir. Per això cal que bàsicament es dediqui a tornar a ocultar els elements visibles en la

interfície en el moment en que el botó és clicat, de tal manera que tornem a l'inici, és a dir, en el punt en que només teníem un botó que deia '*ESCULL ARXIU*'.

Per poder fer-ho sense problemes i estalviant feina innecessària, hem col·locat un comptador en cada una de les sis funcions explicades fins ara. Així, quan l'usuari decideixi clicar el botó '*TORNA A COMENÇAR*', el programa es dedicarà a ocultar només els elements que sabem que estan a la vista i que ho sabem mirant en quin punt està el comptador.

Com s'ha mostrat al llarg de la explicació de la interfície gràfica, els elements que la conformen es van fent visibles mica en mica, i per tant, si cada vegada que fem visible algun nou element augmentem un comptador, podrem controlar exactament quants i quins elements estaran visibles en qualsevol moment que ho vulguem saber. Així doncs, un cop activada la opció de '*TORNA A COMENÇAR*' i per tant la funció '*neteja*', el primer que es farà serà comprovar el valor del comptador, i depenent de quin sigui, amagarem uns elements o un altres. Sempre seran els visibles en aquell moment, ja que sabrem quins encara no s'han fet visibles i que per tant no cal ocultar.

Així doncs, sigui quin sigui el moment en que l'usuari decideixi tornar a començar, la aparença de la interfície que en resultarà després de clicar el botó '*TORNA A COMENÇAR*' serà la següent.

Figura5.8

Constatem que quan cliquem a '*TORNA A COMENÇAR*', realment tornem a començar el procés des de l'inici.

6 Proves

Mostrarem vàries pantalles de la interfície gràfica, un cop executat tot el sistema encriptador/desencriptador.

Podrem observar, sobretot mitjançant l'espectre dels tres senyals representats (original, encriptat i recuperat, respectivament), com ha afectat la encriptació i la desencriptació al senyal d'entrada. Podrem també constatar que si els dos paràmetres configurables per l'usuari, és a dir, el nombre de subbandes i la clau d'encriptació, no coincideixen per l'execució de l'encriptador i per l'execució del desencriptador, els resultats de recomposició en cap cas seran el senyal original. Es mantindrà la encriptació sobre aquest.

Mostrarem primer execucions efectuades amb un arxiu de veu anomenat '*hombre.wav*' en el que se sent una frase dita per un home. En veurem tres exemples, l'un correcte, i els altres dos amb una de les claus errònies en cada cas.

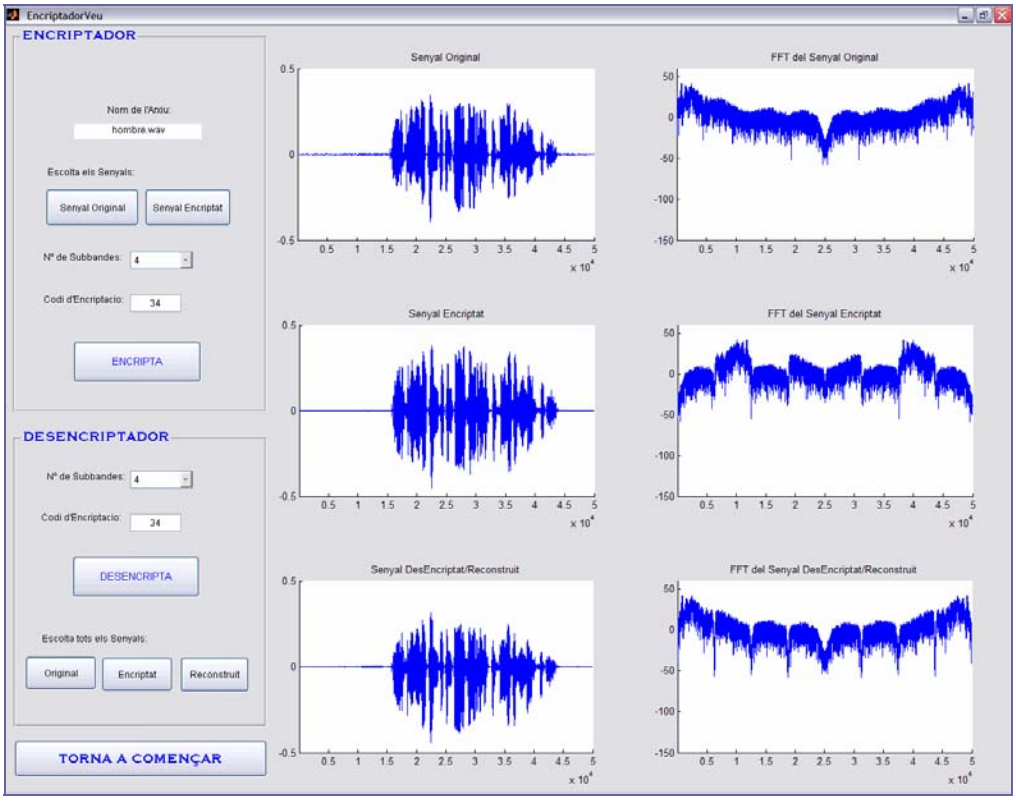

Figura6.1

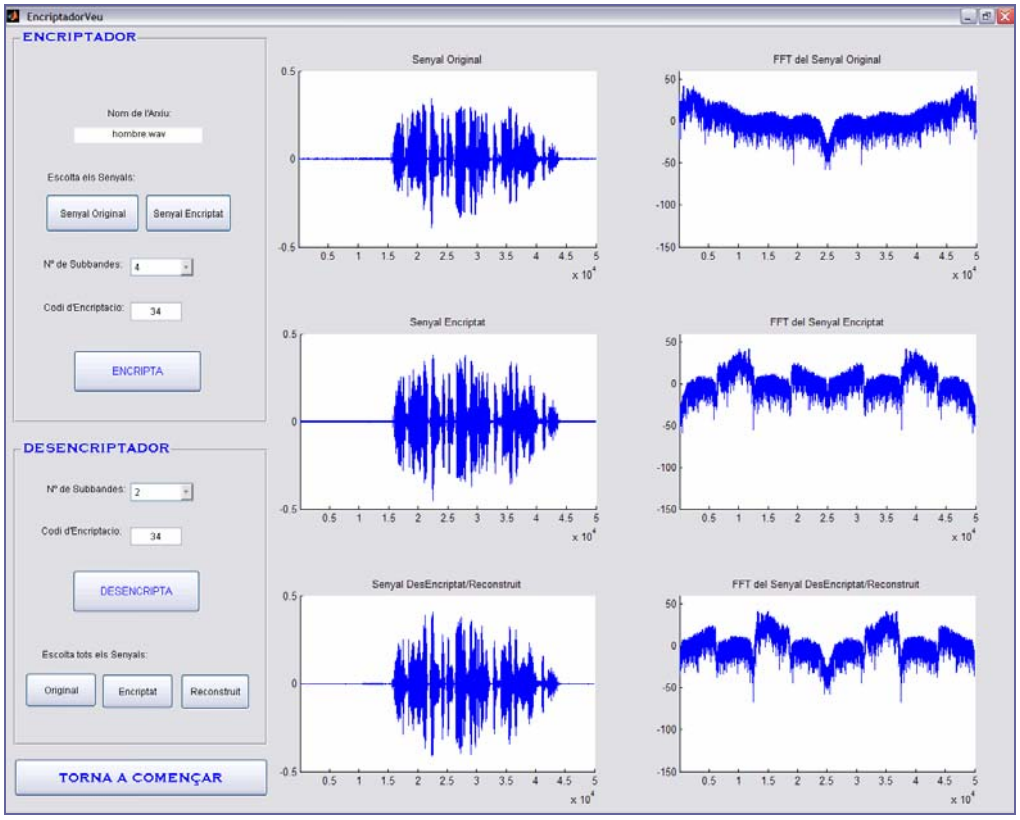

Figura6.2

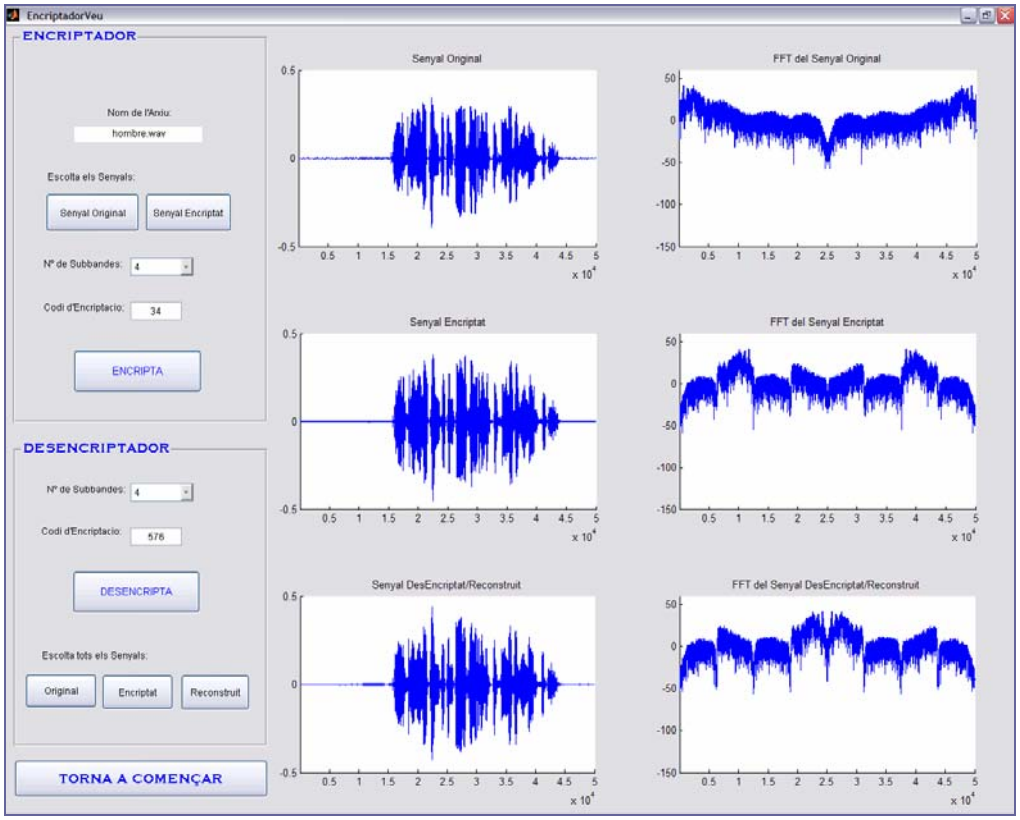

Figura6.3

Ara veurem els mateixos tres resultats obtinguts amb l'execució del procés per a l'arxiu anomenat '*destruct.wav*', on hi ha una espècie de compte enrere dit per un home, amb certa distorsió.

En aquests tres casos, igual que els anteriors, apreciarem tres execucions diferents. La primera on les claus coincideixen i per tant la desencriptació en retorna el senyal original, i les altres dues on una de les dues claus no és la correcta, en un cas el nombre de subbandes i en l'altre la clau d'encriptació.

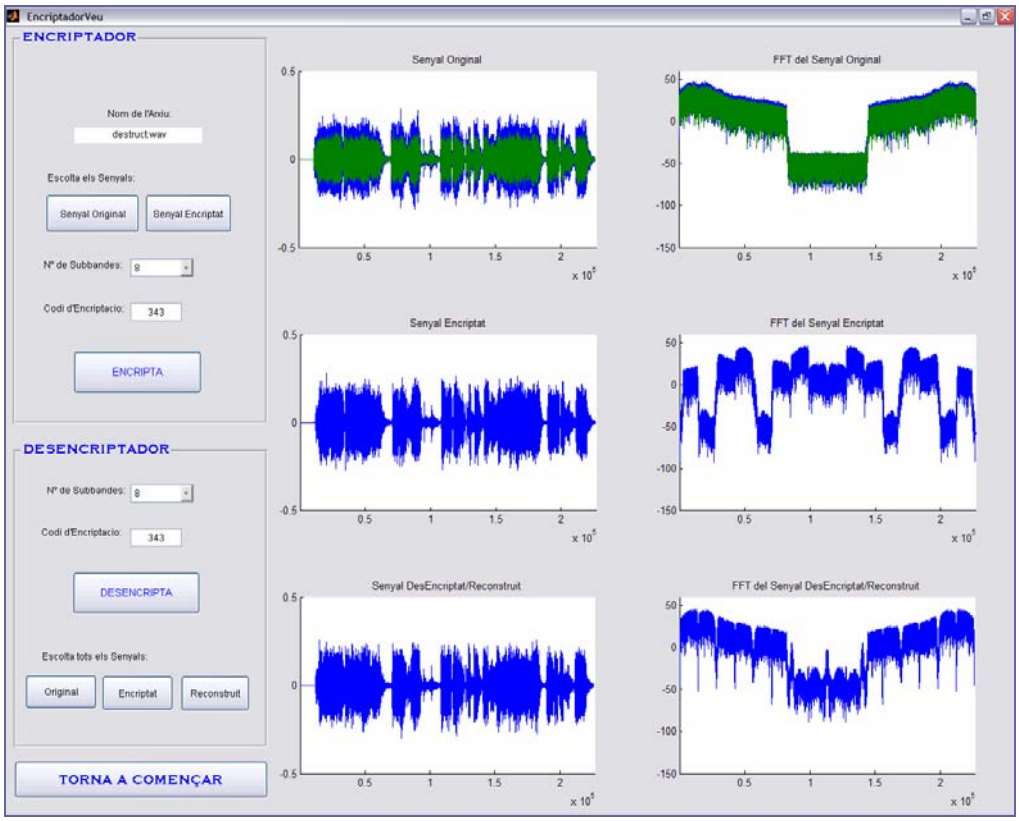

Figura6.4

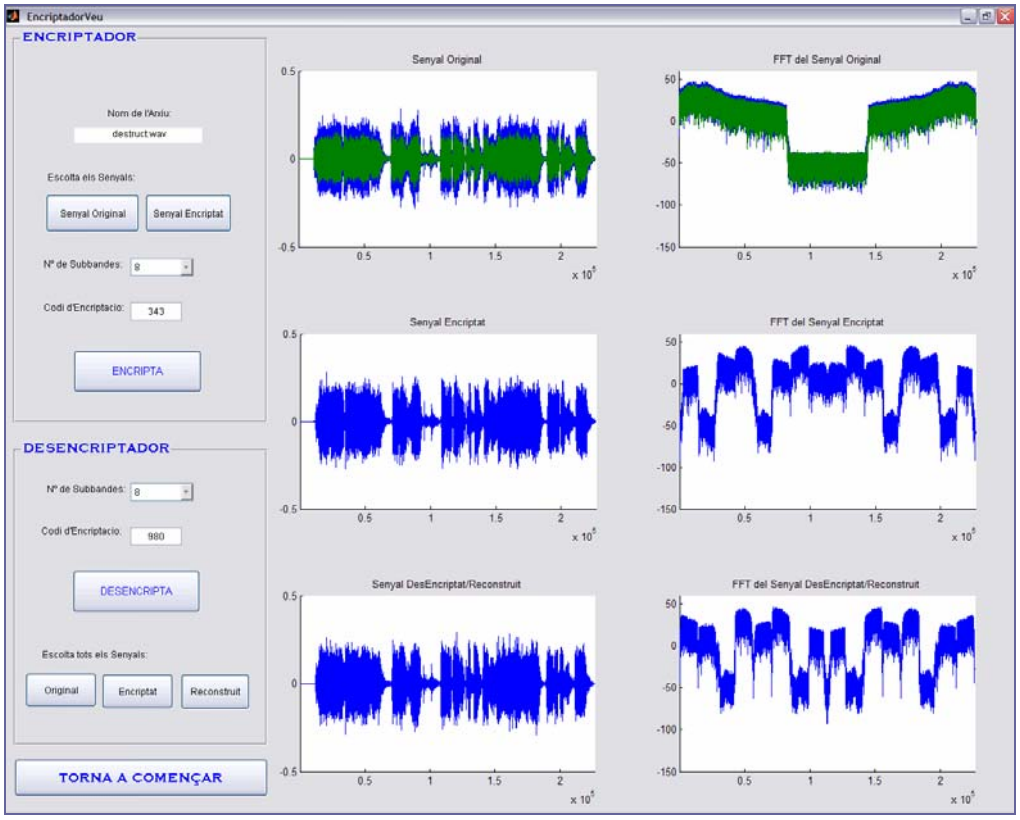

Figura6.5

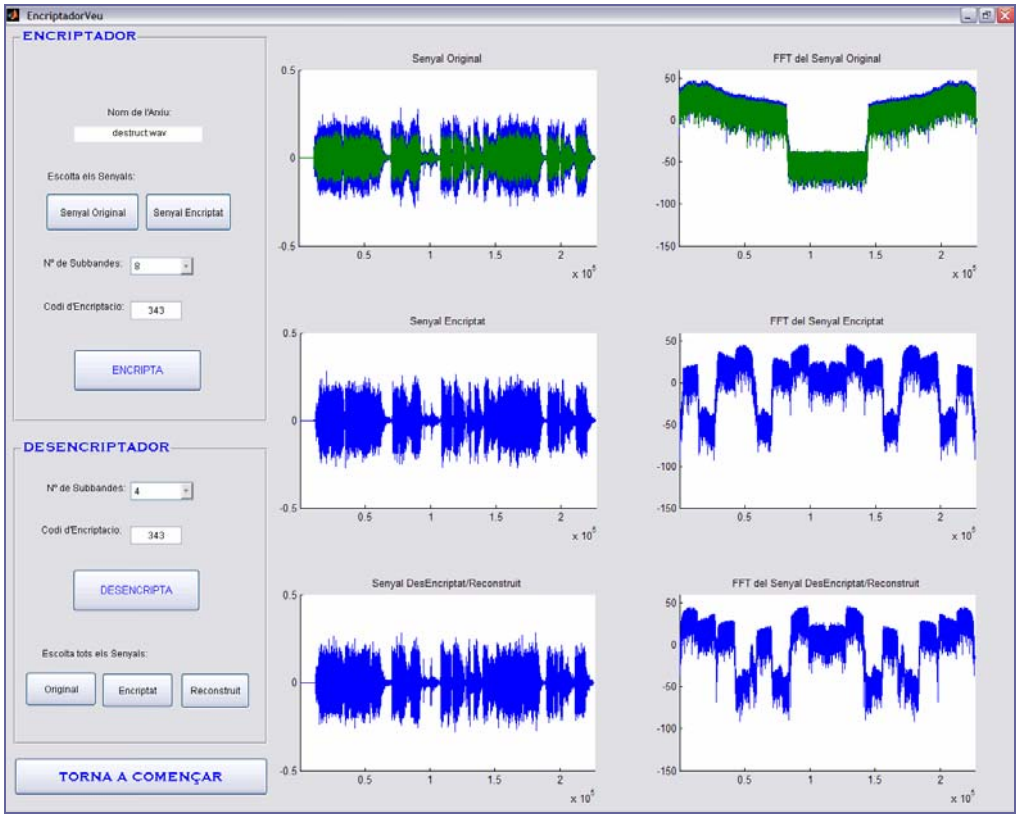

Figura6.6

Veurem els resultats per a l'arxiu '*conixiua.wav*', on trobem un nen japonès saludant. Aquest cop en veiem quatre casos diferents.

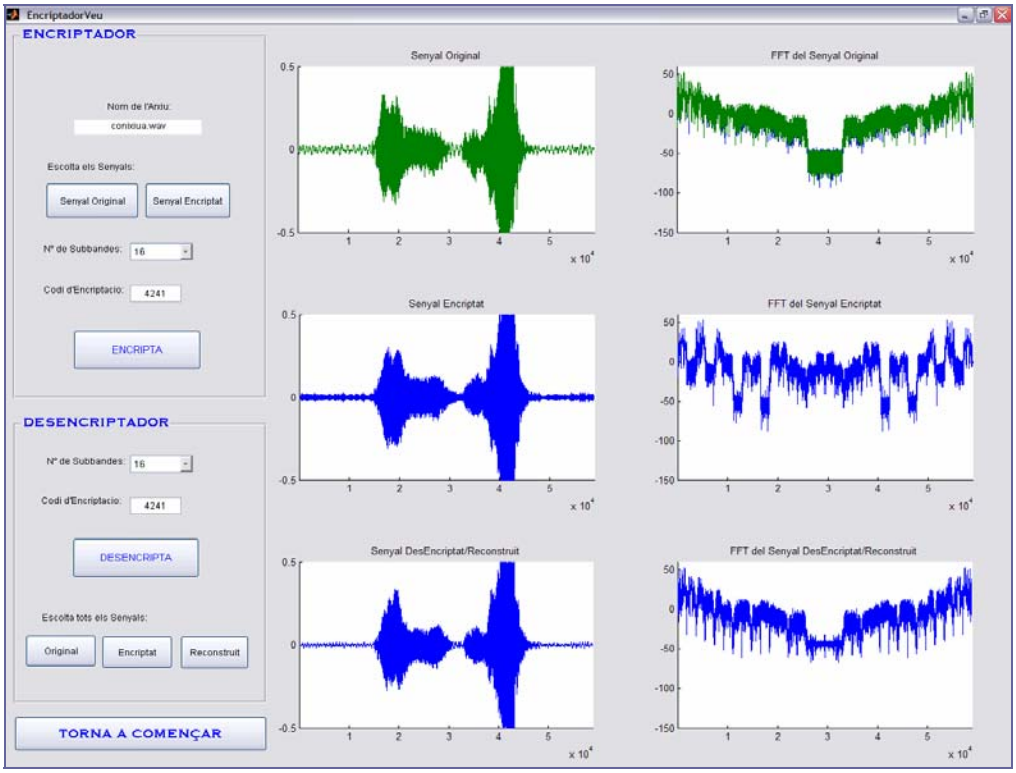

Figura6.7

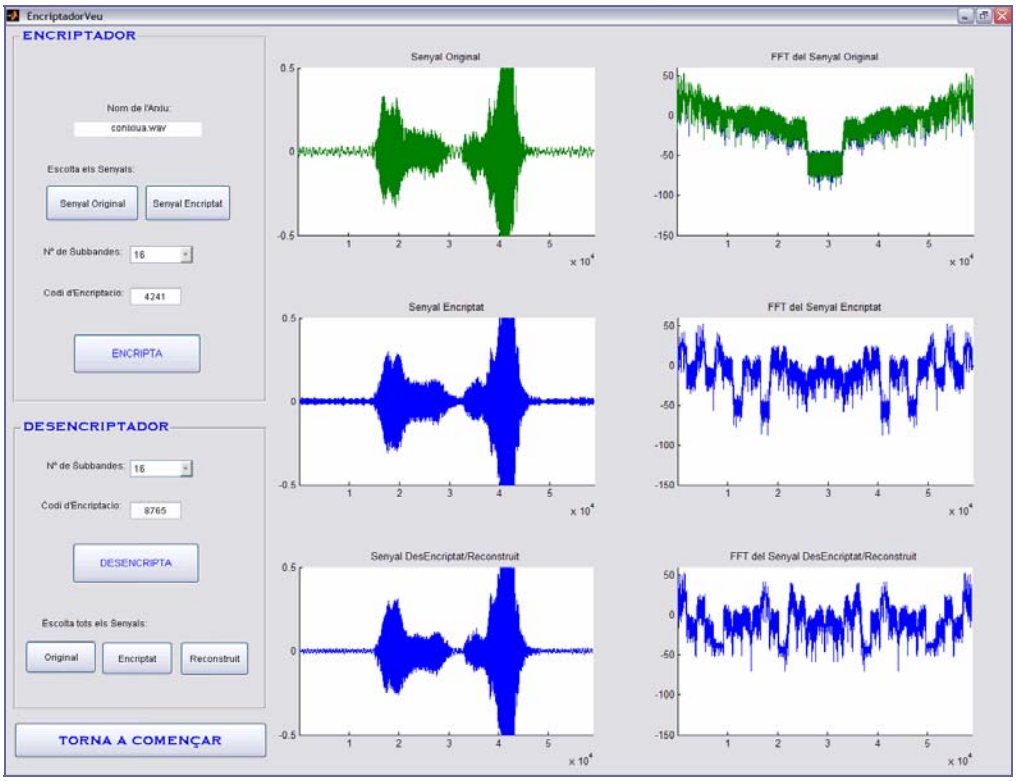

Figura6.8

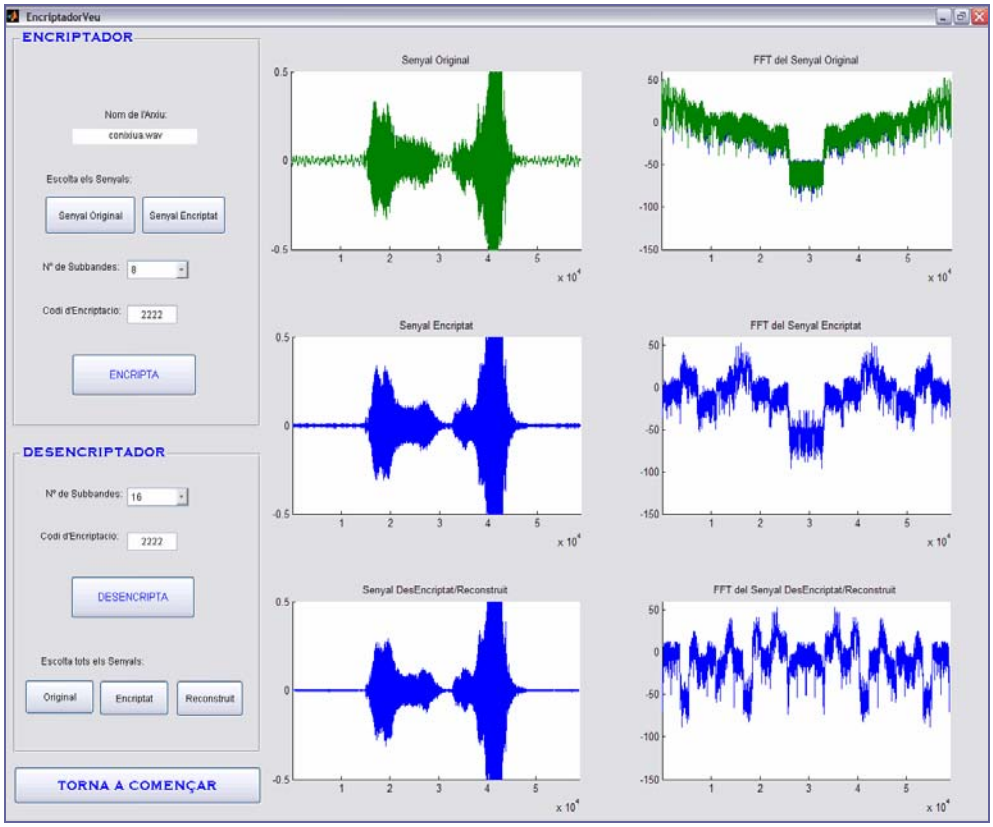

Figura6.9

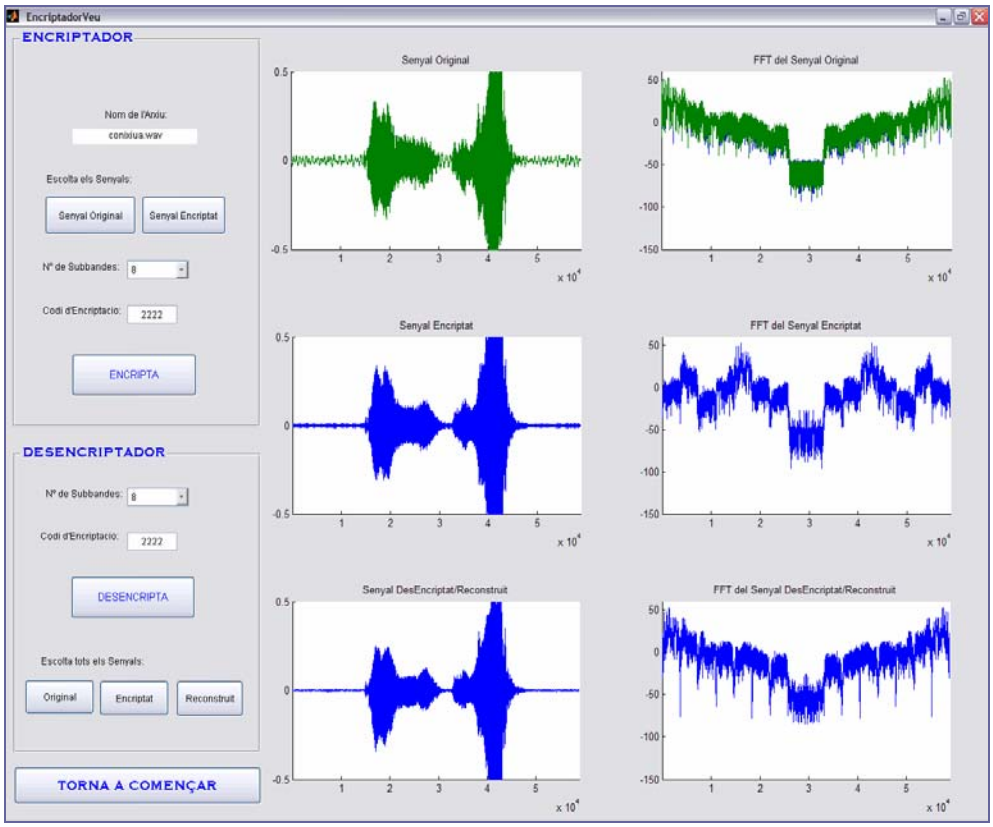

Figura6.10

Finalment, mostrarem una execució realitzada amb un arxiu de música, per comprovar que el nostre sistema també és útil per encriptació d'àudio, no només veu. Concretament és un fragment d'una cançó d'estil 'reggae', amb molta varietat d'instruments de vent i rítmics, on veurem com es descompon i es torna a compondre, perdent molt poca informació del senyal.

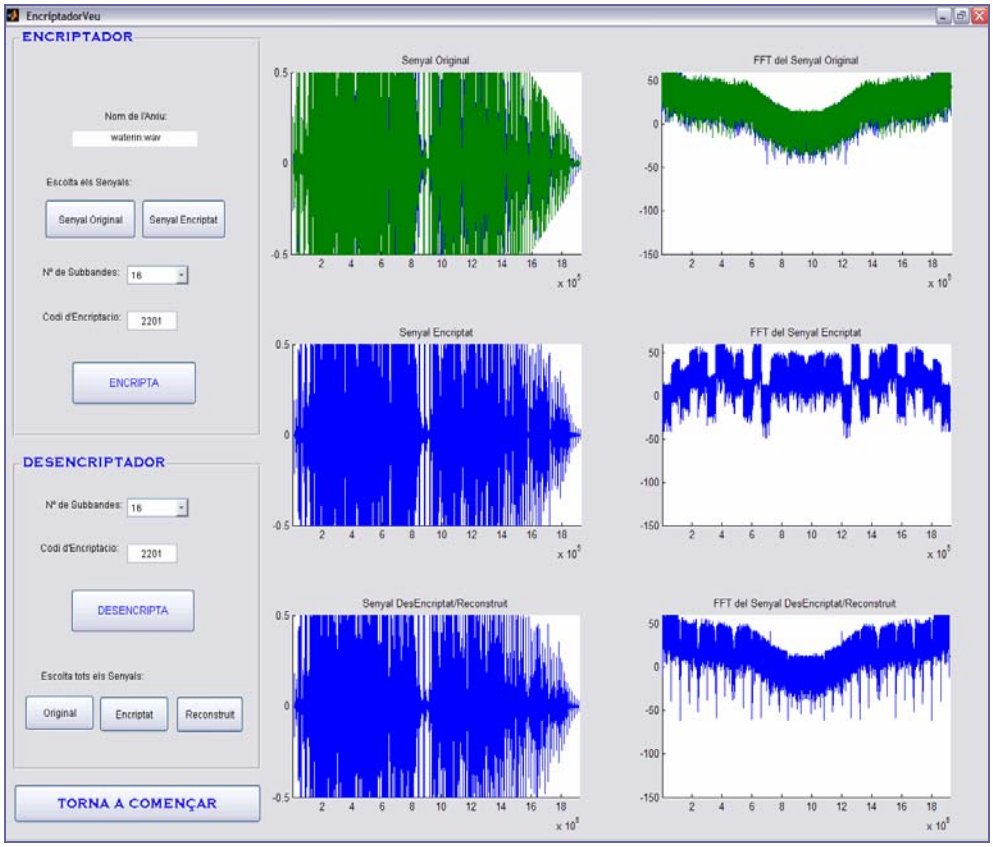

Figura6.11

Amb aquest últim exemple i tots els altres exposats, es demostra que el sistema d'encriptació/desencriptació que s'ha creat funciona com s'esperava, no només alhora de fragmentar i recuperar el senyal amb poques pèrdues d'informació, si no que també queda clar que si els dos paràmetres de configuració 'secrets' usats en desencriptació no coincideixen amb els usats en encriptació, no hi ha manera de recuperar el senyal original. Per molt que un dels dos paràmetres coincideixi, no n'hi ha prou. Necessitem saber els dos exactament.

7 Conclusions i possibles millores

7 1 Conclusions

En aquest punt veurem si els objectius que ens havíem marcat a l'inici del projecte s'han assolit, examinant la feina feta al llarg d'aquest any de treball i valorant-ne els resultats, tant numèrics com personals.

Així doncs farem tres valoracions: el funcionament del sistema implementat, els coneixements assolits, i finalment, la valoració personal de la experiència.

La implementació del sistema encriptador/desencriptador de veu ha resultat ser molt eficaç. S'ha aconseguit crear un sistema d'encriptació simètric, de clau secreta, capaç d'alterar la intel·ligibilitat del senyal en bona mesura, i alhora capaç de recuperar aquesta intel·ligibilitat sense unes pèrdues d'informació gaire grans. Tot això complint el requisit d'un sistema de clau secreta, és a dir, que qui no tingui accés als dos paràmetres privats de configuració del procés d'encriptació (nombre de subbandes i clau d'encriptació privada), no podrà pas recuperar el senyal original. Sempre el continuarà tenint encriptat i per tant inintel·ligible.

També s'ha aconseguit crear una interfície gràfica molt completa. Amb això es vol dir que la interfície creada és un medi d'execució del sistema encriptador/desencriptador idoni, ja que permet que qualsevol usuari, amb coneixements previs o no, entengui què està passant, tant per la presentació adequada dels resultats (visuals i sonors) que permeten veure l'efecte del sistema sobre el senyal introduït, com per la possibilitat de que l'usuari intervingui activament en el procés i comprovi quins són els paràmetres que es poden configurar, i quina repercussió tenen en el procés.

Pel que fa a coneixements assolits, s'ha de dir que realment és molt gratificant quan a partir d'un projecte d'aquest tipus, en el que resulta que has d'endinsar-te en varis temes estudiats prèviament durant la carrera, és ara quan trobes un sentit i unes utilitats a coses que fins aquest moment eren simples temaris d'assignatures, sense anar més enllà. Com ja dic, he après moltes coses que fins ara em sonaven, però que ara han pres un sentit teòric i pràctic molt més interessant, al meu parer. Un exemple podria ser el mateix programa Matlab. Un cop acabat tot el procés de programació, tant del sistema encriptador/desencriptador com de la interfície gràfica, es pot dir que he assolit un bon nivell de coneixements d'un programa fins ara conegut però desconegut alhora, i que em proporciona un gran ventall de possibilitats i recursos.

Finalment, la meva valoració personal, que ja he introduït en part en les valoracions acabades d'exposar, és de satisfacció. Satisfacció per l'acabament, al meu parer, molt d'un projecte llarg i laboriós, que m'ha dut més maldecaps dels que esperava, però que alhora m'ha donat grans satisfaccions. M'ha obert una mica més el camp de visió i m'ha permès entendre millor el funcionament de la universitat i dels estudis que fins ara he estat cursant. Trobant-los un sentit que en alguns moments de la carrera no sabia on ni com buscar. Per fi!

7 2 Possibles millores

Un cop acabada tota la feina de programació hem constatat el bon funcionament del sistema, però tampoc hem d'oblidar que és un sistema d'encriptació d'arxius d'àudio, el que ens pot limitar una mica les seves aplicacions.

El que realment seria una millora substancial d'aquest projecte seria implementar-lo per treballar a temps real. En els sistemes de telefonia per exemple, tant mòbil com fixa, seria bo poder tenir accés a un sistema com

aquest, però amb funcionament a temps real, ja que seria una bona eina alhora de conservar la confidencialitat de converses.

També es podria millorar si el sistema treballés amb arxius del tipus 'mp3', no només 'wav'. Els 'mp3' són de tamany considerablement més reduït i un sistema com el que s'ha creat que els pogués encriptar i desencriptar, permetria intercanvi d'aquest tipus d'arxius confidencialment. Seria aplicable a la distribució de música per internet, legalment per part de les discogràfiques, o a l'intercanvi d'arxius entre usuaris restringits d'un grup de treball, etc. La complicació que ens presenta el sistema no és pas la encriptació o la desencriptació, que es podrien dur a terme idènticament a com s'ha implementat, si no que s'hauria de crear una funció de lectura d'arxius 'mp3', ja que el Matlab no en té cap de predefinida.

Una altra possible millora seria el canvi d'estructura alhora de descompondre el senyal en subbandes. Seria bo implementar-lo de tal manera que la descomposició la féssim sempre sobre la banda de baixes freqüències obtinguda a partir de la prèvia descomposició.

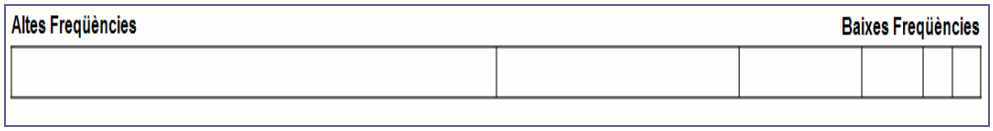

Figura7.1

La major descomposició a baixes freqüències pot permetre una millor encriptació, ja que és en aquesta zona de l'espectre on trobem major informació del senyal, i per tant, si anem fragmentant el senyal mica en mica, però sempre sobre les baixes freqüències, la permutació de les subbandes que se n'obtinguin donarà una alteració de la intel·ligibilitat més elevada, que és el que es busca en un sistema d'encriptació d'aquest tipus. I amb la posterior desencriptació podríem recuperar de nou el senyal original, amb la mateixa taxa de pèrdua d'informació que aquest sistema que s'ha creat ja ens dóna.

8 Bibliografia

Llibres

- M.Faúndez, "*Tratamiento digital de voz e imagen y aplicación a la multimedia*", Marcombo Baixareu Editores, 2000

- B.Gold i N.Morgan, "*Speech and Audio Signal Processing*", Wiley Press, 1999

- V.K.Ingle i J.G.Proakis, "*Digital Signal Processing Using Matlab*", Prentice Hall, 1996

- S.W.Smith, "*The Scientist and Engineer's Guide to Digital Signal Processing*", California Technical Publishing, 1997

- A.Carrión, "*Acústica. Apunts*", Departament TSC - EUETIT, 2003

- A.Carrión, "*Acústica. Transparències*", Departament TSC - EUETIT, 2003

- A.J.Mendez, P.C.van Oorschot i S.A.Vanstone, "*Handbook of Applied Cryptography*", CRC Press, 1996

- J.Zhou, M.Yung i Y.Han, "*Applied Cryptography and Network Security*", Springer, 2000

Pàgines web

- http://www.mathworks.com - "*The MathWorks. MATLAB and Simulink for Technical Computing*"

- http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.html - "*Documentation for MathWorks Products. MATLAB including External Interfaces/API, GUIDE, Handle Graphics, File I/O, Notebook.*"

- http://es.wikipedia.org/wiki/Portada - "*Wikipedia. La enciclopedia libre. Español*"

- http://www.inicia.es/de/alt64/articulo - "*Artículos en @lt+64*"

- http://www.technologyreview.com - "*Technology Review. The Impact of Emerging Technologies. From MIT. Information on Emerging Technologies & impact on business & society.* "

- http://www.technologyreview.com/channel/info.aspx - "*Technology Review. Infotech Channel Homepage*"

- http://www.iec.es - "*Institut d'Estudis Catalans*"

- http://www2.iecat.net/gc/...
.../SECCI%26Oacute%3B+FILOL%26Ograve%3BGICA - "*Institut d'Estudis Catalans. Secció Filològica"*

- http://www.whatis.com - http://whatis.techtarget.com - "*Wahtis?com. The learning IT encyclopedia and learning center*"

- http://www.searchsecurity.com - http://searchsecurity.techtarget.com - "*SearchSecurity.com. The Web's best security-specific information resource for enterprise IT professionals*"

Agraïments

Vull agrair primer de tot a la família i als amics (que no anomenaré, els aludits ja sabeu qui sou!), per la paciència i l'ajuda, més que res moral, que m'han donat en aquest últim any de carrera. Han hagut d'aguantar molts maldecaps per culpa d'un projecte (aquest que teniu a les mans) que, sincerament, poc han entès... qui no està ficat una mica en aquest món, li sona a xino tot això! Però tot i així, m'han sabut aguantar i fins i tot algú hi ha mostrat interès. Per tot això, i molt més, moltes gràcies a tots!

També vull donar les gràcies a l'Ignasi, el tutor d'aquest projecte, per la planificació i l'ajuda que m'ha proporcionat. Potser hem estat més temps del que tocava en això, però estic molt content del resultat obtingut. Ha valgut la pena, tant acadèmica com personalment. I he descobert noves facetes de qui fins ara havia estat simplement un professor més. Gràcies.

Vull agrair, i de forma molt efusiva (la més efusiva de totes!), a les terres gironines, que amaguen gent tant genial, i única! Què hagués fet sense el Shiatsu i les nits de dimecres?! No m'ho vull ni imaginar! Gràcies de TOT cor!!

Per acabar, gràcies a tothom que hagi arribat fins a aquesta pàgina de la memòria...vol dir que ha trobat interès en algo que hi he dedicat molts esforços, i això em gratifica enormement! *Grazie mile*!!

Annex1: Codi informàtic del sistema encriptador/desencriptador

Annex2: Codi informàtic de la interfície gràfica